NEW TOOLS FOR UNSUPERVISED LEARNING

A Thesis
Presented to
The Academic Faculty

by

Ying Xiao

In Partial Fulfillment
of the Requirements for the Degree
Doctor of Philosophy in Algorithms, Combinatorics and Optimisation in the
School of Computer Science

Georgia Institute of Technology
December 2014

NEW TOOLS FOR UNSUPERVISED LEARNING

Approved by:

Santosh Vempala, Advisor
School of Computer Science
Georgia Institute of Technology

Maria Florina Balcan
School of Computer Science
Carnegie Mellon University

Justin Romberg
School of Electrical and Computer Engineering
Georgia Institute of Technology

Le Song
Computational Science and Engineering
Georgia Institute of Technology

Arkadi Nemirovski
School of Industrial and Systems Engineering
Georgia Institute of Technology

Date Approved: 4 August 2014

To my parents, Qiong Huang and Yongshun Xiao.

ACKNOWLEDGEMENTS

Deep are the debts of gratitude owed to those around us: for what man are they not infinite?

My deepest thanks are certainly owed to my advisor, Santosh Vempala. He is a man of considerable ability and character; I have learnt much from him, and not solely in the endeavour of academic research.

I am very grateful to the ACO program at Georgia Tech, ably led by Robin Thomas. This has been a formative experience, and has provided excellent training for whatever comes next. I thank my collaborators Lev Reyzin, Elena Grigorescu, Vitaly Feldman and Navin Goyal for all the efforts that they have put into our work. Without their insight and work, this thesis would hardly have been possible, and in any case the experience would have been vastly poorer.

I'd like to thank all the friends I've made in Atlanta over the last five years. My erstwhile room-mates Brendan Dolan-Gavitt, Catherine Grevet, Hank Carter and Anita Zakrzewska have all shared their lives with me, and it has been such an incredible privilege. My thanks to my close friends Karthik Raveendran, Mark Luffel, Ben Cousins, Chris Berlind and Emma Cohen – this would have been a duller existence and less rewarding experience without you. My friend Prateek has been, in particular, a font of mathematical wisdom and loyal friendship. Finally, a heartfelt thank you to Shawn Dyer, and especially Tara Mooney, for reminding me that there exist worlds beyond mathematics!

TABLE OF CONTENTS

SUMMARY

In an unsupervised learning problem, one is given an unlabelled dataset and hopes to find some hidden structure; the prototypical example is clustering similar data. Such problems often arise in machine learning and statistics, but also in signal processing, theoretical computer science, and any number of quantitative scientific fields. The distinguishing feature of unsupervised learning is that there are no privileged variables or labels which are particularly informative, and thus the greatest challenge is often to differentiate between what is relevant or irrelevant in any particular dataset or problem.

In the course of this thesis, we study a number of problems which span the breadth of unsupervised learning. We make progress in Gaussian mixtures, independent component analysis (where we solve the open problem of underdetermined ICA), and we formulate and solve a feature selection/dimension reduction model. Throughout, our goal is to give finite sample complexity bounds for our algorithms – these are essentially the strongest type of quantitative bound that one can prove for such algorithms. Some of our algorithmic techniques turn out to be very efficient in practice as well.

Our major technical tool is tensor spectral decomposition: tensors are generalisations of matrices, and often allow access to the "fine structure" of data. Thus, they are often the right tools for unravelling the hidden structure in an unsupervised learning setting. However, naive generalisations of matrix algorithms to tensors run into NP-hardness results almost immediately, and thus to solve our problems, we are obliged to develop two new tensor decompositions (with robust analyses) from scratch. Both of these decompositions are polynomial time, and can be viewed as efficient generalisations of PCA extended to tensors.

CHAPTER I

INTRODUCTION

In an unsupervised learning problem, one is given an unlabelled dataset and hopes to find some hidden structure; the prototypical example is clustering data, but unsupervised learning also encompasses dimensionality reduction, feature selection, and a number of latent variable models. Such problems arise in a variety of settings – naturally in machine learning and statistics, but also in signals processing, theoretical computer science, and any number of quantitative scientific fields. The distinguishing feature of unsupervised learning is that there are no privileged variables or labels which are particularly informative, and thus the greatest challenge is often to differentiate between what is relevant and what is irrelevant in any particular dataset or problem. Perhaps the best-known technique in unsupervised learning is Principal Component Analysis (PCA) where one computes the singular value decomposition of the data matrix.

Unsupervised learning is ubiquitous in practice, yet our theoretical understanding is relatively immature. In this thesis, we move to address certain deficiencies in our theoretical understanding: we carefully define a number of unsupervised learning problems, and then give algorithms that provably solve them, being economical in our use of computation and samples. The problems we select are well-known in the literature – independent component analysis (for which we solve the open underdetermined case, as well as give an extremely efficient algorithm for the easier fully determined case), mixtures of spherical Gaussians (where we match the state of the art), and a dimensionality reduction or feature selection model (where our contribution is partly the specification of the model, partly the general algorithmic framework, and partly the applications).

In all the above cases, we are interested in studying the accuracy and efficiency of our algorithms in the realisable case, that is, given a finite (but growing) number of samples

drawn from the ideal distribution. Finite sample complexity is strictly stronger than statistical convergence (wherein it is only required to achieve accuracy in the limit). Moreover, our notion of efficiency will be the familiar requirement of polynomial run-times for our algorithms; this is stronger than, for example, the notion of computational convergence wherein one only requires the algorithm to stabilise to an answer in the limit). Thus, one can view finite sample complexity and polynomial runtime as quantitative statements of consistency and convergence.

The tool which underlies our learning results is tensor decomposition; tensors are generalisations of matrices, and as such are able to overcome many of the barriers and limitations associated with matrix methods. Roughly speaking, tensors often give a good understanding of the "higher order information" about the data, and thus are often the right tools for unravelling the hidden structure in unsupervised learning setting. However, the algorithmic theory of tensors is quite immature and little is known about provably good algorithms. In this thesis, we develop two new tensor decomposition techniques, and also prove some new structural theorems about tensors. The holy grail in the study of tensors is to develop a general purpose PCA-like tool for tensors, but unfortunately most natural generalisations of PCA run into computational hardness results; thus, the recurring motif of this thesis is the need to dodge this fundamental NP-hardness. To attain the finite sample complexity bounds, we must make our tensor decompositions robust and efficient, and much of the technical work is in proving these two properties for our decompositions. After developing the necessary tensor tools – both structural results and algorithmic techniques – we apply these to our unsupervised learning problems. Let us now briefly discuss these problems.

In an *Independent Component Analysis* (ICA) problem, one is given independent samples $x \in \mathbb{R}^n$ given by the form:

$$x = As$$

where $A \in \mathbb{R}^{n \times m}$ is an unknown fixed linear map and $s \in \mathbb{R}^m$ is a fully independent random vector with unknown distribution. We think of the variables x as observed variables, s as latent variables and A as a "mixing matrix" that gives a linear relationship between the

two. The problem is to compute the mixing matrix A from a sequence of independent and identical (i.i.d.) samples $x^1, x^2, \ldots$ of observation variables. This problem has a long and distinguished history (for example, [76]), and is a prominent model that crops up in machine learning, statistics and signal processinging. ICA is used in all sorts of practical applications as a dimensionality reduction tool, and most recently has found a place in sparsifying layers of deep belief networks [106].

Note that *a priori*, we assume no relationship between the dimension of the observed variables n and the dimension of the latent variabes m; in doing so, this leads to three regimes: when $n > m$, the problem is overdetermined; when $n < m$, the problem is termed underdetermined; of course when $n = m$ the problem is fully determined. In terms of algorithms with provably good finite sample complexity, the only results in the literature are for the fully determined case when $n = m$ [62, 107, 15, 21, 9]. It is also well known, as a folklore result, that a simple application of PCA suffices to convert the overdetermined problem to a fully determined. The major contribution of this thesis towards ICA is the design and analysis of an algorithm for the underdetermined ICA problem. For the fully determined case, we give an extremely efficient algorithm that works both in theory and practice.

The major component of this algorithm is a robust tensor decomposition that we apply to higher order derivatives of the Fourier transform of the distribution computed at randomly chosen points. Roughly speaking, we develop an algorithm that allows us to spectrally decompose rank 1 tensors which are suitably nice (in much the same way that eigenvectors give a spectral decomposition for matrices). Computing such a decomposition for a single tensor is NP-hard [72], but our structural property sidesteps this computational hardness; for an order r tensor, we are able to decompose tensors with with $\binom{n}{r}$ rank 1 components. There has already been subsequent work in applying our technique to clustering problems and other unsupervised learning problems [13].

A second problem to which we can apply this tensor technique is the separation of certain types of Gaussian mixtures. Gaussian mixture models are a clustering model where one postulates that the data is drawn from a set of k unknown (but fixed) Gaussian distributions.

Gaussian distributions are relatively localized in space, and thus one can think of the points generated by any particular Gaussian as a cluster. This is in fact one of the oldest statistical models, with its origns tracing back to a paper of Pearson from 1894 [110]. This problem is of central importance, and is probably the most applied clustering technique after k-means. Much is already known about this problem – the state-of-the-art is essentially able to recover the parameters of the mixtures of k arbitrary Gaussians over $\mathbb{R}^n$ in time polynomial in n (though not polynomial in k) [101]. Our method for Gaussian mixtures is not formally comparable to this – we are able to recover mixtures of n Gaussians with spherical or isotropic covariances in polynomial in n time. Our work is essentially as strong as [9], and a clever extension of our technique in [13] is able to find more than n Gaussians in n dimensions in polynomial time. The latter work also uses our tensor decomposition.

We also develop a second type of tensor decomposition which we call the additive subspace decomposition. This generalises the notion of a decomposition of a tensor into rank 1 tensors (for matrices, the spectral theorem does this), and sometimes allows us to decompose tensors even when no rank 1 decomposition exists. Our algorithm for this tensor decomposition employs sequential local search/gradient-descent moves and combines with a robust polynomial-identity testing scheme.

We use the method to give an algorithm for dimensionality reduction and finding relevant subspaces in learning problems. The motivation is to generalise the idea of feature selection. To prevent overfitting in supervised learning problems (i.e., classification or regression) one will often run a feature selection algorithm to reduce the dimensionality of the feature space. The intuition is that there exists some set of relevant features and the remaining features are irrelevant to the prediction of the privileged variable. Oftentimes, these feature selection algorithms are supervised, and use the response variable to guide the process, but the more common approach is to attempt an unsupervised dimensionality reduction using, for example, PCA, random projection or ICA. A formalisation of this idea in learning theory is known as the *junta* model, where there are a small number of relevant features and and a large number of irrelevant features, and the task is to sift through all of them and identify which is which. This type of model is rather strong as it implicitly assumes that we know

the right basis for the data, but in many signal processinging situations, for example, the measurements are in fact linear combinations of some underlying subset of variables.

Combining these ideas, we define the *subspace junta* model: here the relevant variables live in some unknown, fixed dimensional subspace of the entire feature space. In the orthogonal irrelevant subspace, we think of the irrevelant features as being totally unrelated to the response variable, and in fact in our strongest variant, the features simply take on values given by Gaussian measurement error. Thus, we can write the overall distribution F over $\mathbb{R}^n$ as $F = F_V F_{V^\perp}$ i.e., a product of two independent marginals F_V over the subspace V and $F_{V^\perp}$ over the orthogonal subspace $V^\perp$.

To tackle the problem of finding the relevant subspace, we give a local-search based tensor algorithm for finding the relevant subspace, which exploits the independence assumptions. Our algorithm relies on the moment tensor of the overall distribution: the main algorithmic step is to solve an optimisation problem associated with the moment tensor (for matrices, this amounts to PCA or computing the top eigenvalue of the covariance matrix). Unfortunately, the solution of this optimisation problem for higher moments is once again NP-hard; to side-step this hardness, we exploit the local optima of this optimisation problem, rather than the global optimum. These local optima are easy to compute, and give enough information to reconstruct the relevant subspace. Once again, one can view the computation of these local optima as tensor generalisations of the spectral decomposition for matrices.

Finally, in work outside of this thesis, we give a sample complexity bound for the planted clique problem – a toy clustering model. We refer the interested reader to [60]. In particular, this paper highlights the limitations of our tensor driven approach.

We will now proceed to give the necessary background, first in unsupervised learning in Chapter 2, with particular care devoted to the successes and limitations of PCA. Following this, we will describe our tensor algorithms in Chapter 4 – the exposition will be self-contained and will be disjoint from machine learning. We will finally apply these techniques to unsupervised learning problems in Chapter 5.

CHAPTER II

UNSUPERVISED LEARNING: MODELS AND PROBLEMS

2.1 Introduction

In this chapter, we give a detailed description of the unsupervised learning problems that we study in this thesis. In an unsupervised problem, one is given a set of unlabelled data (represented by m rows of data, each of which lives in $\mathbb{R}^n$ say), and one attempts to find a hidden or latent structure in the data. This is in contrast with the usual supervised learning, where one is given labelled data that is in addition to the data matrix which lives in $\mathbb{R}^{m \times n}$, one is also given a special privileged variable per row of data, and the task is to predict this response variable on future samples.

Unsupervised learning encompasses a vast host of problems – clustering, latent variable models and dimensionality reduction all fall under its umbrella (see also the standard reference [70]). During the course of this thesis, we study problems from all three of these domains. Here we describe these problems in more detail and provide the relevant background and definitions.

2.2 Independent component analysis

The first unsupervised learning problem is a specific latent variable model called Independent Component Analysis (ICA) [80].

Problem 1 (ICA)**.** *Let $n, m \in \mathbb{N}$. We say that $x \in \mathbb{R}^n$ is generated by an ICA model if $x = As$ where $A \in \mathbb{R}^{n \times m}$ is some fixed but unknown matrix, and $s \in \mathbb{R}^m$ is a fully independent random vector. The problem is to recover the columns of A from independent samples $\{x^1, x^2, \ldots\}$ up to symmetries.*

As an additional assumption, we require that none of the s_i are Gaussian for the simple reason that Gaussian random vectors fail to be unique up to unitary transformations, thus, one cannot hope to recover A in case more than one of the s_i are Gaussian. As an interesting

side-effect, all ICA algorithms must require that the component distributions differ from being Gaussian in some fashion.

Special cases of ICA are of interest in many application areas with large or high-dimensional data sets [76]. Jutten and Herault formalized the ICA problem [80] and mention in their paper that variants of this problem had appeared in a variety of different fields prior to this (the earliest such mention is in [17]). This type of "blind source separation" or "deconvolution" problem is prevalent in diverse areas ranging from signal processing to neuroscience to machine learning.

The notion that random variables should be far from Gaussian pervades ICA research. By the central limit theorem, sums of independent random variables converge to a Gaussian, whereas individually the latent random variables are not Gaussian. Thus finding unit vectors that maximize some notion of non-Gaussianity might reveal the latent variables. This intuition is formalized by introducing functions which serve as a proxy for non-Gaussianity, called "contrast functions" in the ICA literature. The definition of a contrast function is that maximizing a contrast function will give an independent component. Some examples of contrast functions include the kurtosis (4th order analogue of variance)[91, 44], various cumulants, and functions based on the so-called *negentropy* ([45]). A number of algorithms have been devised in the ICA community using different contrast functions. The literature is vast and we refer to [47] for a comprehensive account. The ICA problem has been studied rigorously in theoretical computer science in several previous papers [62, 107, 15, 21, 9]. All of these algorithms either assume that the component distribution is a very specific one [107, 15], or assume that its kurtosis (fourth cumulant) is bounded away from 0, in effect assuming that its fourth moment is bounded away from that of a Gaussian. The application of tensor decomposition to ICA has its origins in work by Cardoso [37], and similar ideas were later discovered by Chang [40] in the context of phylogenetic reconstruction and developed further in several works, e.g. Mossel and Roch [104], Anandkumar et al. [8], Hsu and Kakade [73] for various latent variable models. Arora et al. [15] and Belkin et al. [21] show how to make the algorithm resistant to unknown Gaussian noise. Additionally, there are a variety of tensor methods and maximum likelihood methods used [36, 24]. Many of

these methods require that the latent random variables s_i differ from a Gaussian in the same fixed moment (typically fourth); this is a rather inconvenient requirement that we should not expect to hold *a prior*, and we show how to remove it in Chapter 5.

Underdetermined ICA is the hardest form of this problem. In the underdetermined case, there are more independent source variables than there are measurements, thus the mixing matrix A has fewer rows than columns and A is not square or invertible (i.e., it includes a projection). We have to be a little more careful in fixing the normalisation of the columns of A in this case:

Problem 2 (Underdetermined ICA)**.** *Fix $n, m \in \mathbb{N}$ such that $n \leq m$. We say that $x \in \mathbb{R}^n$ is generated by an underdetermined ICA model if $x = As$ for some fixed matrix $A \in \mathbb{R}^{n \times m}$ where A has full row rank and $s \in \mathbb{R}^m$ is an independent random vector. In addition, we fix the normalization so that each column A_i has unit norm. The problem is to recover the columns of A from independent samples x modulo phase factors.*

There are a number of algorithms proposed for underdetermined ICA in the signal processing literature, many of them quite sophisticated. However, none of them is known to have rigourous guarantees on the sample or time complexity, even for special distributions. See e.g. Chapter 9 of [47] for a review of existing algorithms and identifiability conditions for underdetermined ICA. For example, FOOBI [38, 52] uses fourth-order correlations, and its analysis is done only for the *exact* setting without analyzing the robustness of the algorithm when applied to a sample, and bounding the sample and time complexity for a desired level of error. In addition, the known sufficient condition for the success of FOOBI is stronger than ours, more elaborate, and rather more opaque. We mention two other related papers that extend this technique[46, 4]. We also tackle this problem in Chapter 5 and give the first, to our knowledge, provably good algorithm for it. Tensor decomposition techniques, such as power iteration, which are known to work in the fully determined case cannot possibly generalize to the underdetermined case [9], as they require linear independence of the columns of A, which means that they can handle at most n source variables.

A problem related to ICA is learning a dictionary [88, 108], which similarly has found

many applications in signal and image processing. The key difference is that the latent vector s in dictionary learning is given by a sparse random vector supported on at most k components (instead of independent random vectors in ICA). Once again, the goal is to infer the matrix A, which is appropriately called the dictionary matrix – the underlying idea is that the matrix A allows for a favourable, compact encoding of the observed vectors x. As with ICA, the relative dimensions of $A \in \mathbb{R}^{n \times m}$ give rise to three separate interesting regimes. The literature on this problem is enormous, and spans a variety of fields, but we mention here the few provably good algorithms [120, 14, 3, 10, 18]. Although the model substantially resembles ICA, the sparsity of s requires vastly different techniques and assumptions.

2.3 Clustering

In a clustering problem, one is presented with unlabelled data from $\mathbb{R}^n$, and is asked to group them according to some (not necessarily well-defined!) notion of similarity. A natural example of such an approach is to assume that points which are spatially close to each other (e.g., in Euclidean distance) are similar, and thus we want to pick as our groups points which are spatially proximate. One could do this, for example, by minimising the L^2 square norm of the clusters and this would lead to the well-known k-means algorithm. There are many related approaches to clustering whereby one defines an objective function and then tries to pick clusters to minimise this objective function such as k-median, but we will pass over these here as the ideas are generally the same. An alternative approach is known as agglomerative clustering where one grows clusters sequentially by connecting neighbouring points. A dual approach is to recursively divide the data into smaller and smaller portions. A third, more sophisticated, approach is to postulate a latent variable model which captures the cluster, which we will examine in greater detail; we shall also deal with graph clustering.

Gaussian mixture models are a popular model in statistics. A distribution F in $\mathbb{R}^n$ is modeled as a convex combination of unknown Gaussian components. Given i.i.d. samples from F, the goal is to learn all its parameters, i.e., the means, covariances and mixing weights of the components.

Problem 3 (Gaussian Mixtures)**.** *Let $F = \sum_{i=1}^{m} w_i N(\mu_i, \Sigma_i)$ be a distribution over $\mathbb{R}^n$. From iid samples $x^1, x^2, \ldots$ drawn according to F, deduce the parameters $\{w_i, \mu_i, \Sigma_i\}$.*

A classical result in statistics says that Gaussian mixtures with distinct parameters are uniquely identifiable, i.e., as the number of samples goes to infinity, there is a unique decomposition of F into Gaussian components. It has been established that the sample complexity grows exponentially in m, the number of components [23, 22, 81, 101], but only polynomially in n. In a different direction, under assumptions of spatially separable components, a mixture is learnable in time polynomial in all parameters [127, 48, 115, 50, 41, 34].

An even simpler model of clustering is the planted clique problem: we are given a graph G whose edges are generated by starting with an Erdos-Renyi random graph $G_{n,1/2}$ (over n vertices where each edge is independently present with probability $1/2$), then "planting," i.e., adding edges to form a clique on k vertices (which are unknown to us). The goal is to find the location of the clique.

Problem 4 (Planted Clique)**.** *Let $G_{n,1/2}$ denote an Erdos-Renyi random gaph. Suppose that one adds a small complete graph K_k over the (unknown) vertices $\{v_{i_1}, \ldots, v_{i_k}\}$. From the graph $G_{n,1/2} \cup K_k$, deduce the planted clique (i.e., the vertices $\{v_{i_1}, \ldots, v_{i_k}\}$.*

Jerrum [78] and Kučera [90] introduced the planted clique problem as a potentially easier variant of the classical problem of finding the largest clique in a random graph [83]. A random graph $G_{n,1/2}$ contains a clique of size $2 \log n$ with high probability, and a simple greedy algorithm can find one of size $\log n$. Finding cliques of size $(2 - \epsilon) \log n$ is a hard problem for any $\epsilon > 0$. Planting a larger clique should make it easier to find one. The problem of finding the smallest k for which the planted clique can be detected in polynomial time has attracted significant attention. For $k \geq c\sqrt{n \log n}$, simply picking vertices of large degrees suffices [90]. Cliques of size $k = \Omega(\sqrt{n})$ can be found using spectral methods [6, 99, 43], via SDPs [57], nuclear norm minimization [7] or combinatorial methods [58, 54].

While there is no known polynomial-time algorithm that can detect cliques of size below the threshold of $\Omega(\sqrt{n})$, there is a quasipolynomial algorithm for any $k \geq 2 \log n$: enumerate

subsets of size $2 \log n$; for each subset that forms a clique, take all common neighbors of the subset; one of these will be the planted clique. This is also the fastest known algorithm for any $k = O(n^{1/2-\delta})$, where $\delta > 0$.

Some evidence of the hardness of the problem was shown by Jerrum [78] who proved that a specific approach using a Markov chain cannot be efficient for $k = o(\sqrt{n})$. More evidence of hardness is given in [59], where it is shown that Lovász-Schrijver SDP relaxations, which include the SDP used in [57], cannot be used to efficiently find cliques of size $k = o(\sqrt{n})$. The problem has been used to generate cryptographic primitives [79], and as a hardness assumption [5, 71, 100].

2.4 Dimensionality reduction and relevant features

Our third problem of interest is a form of dimensionality reduction. Having data of extremely high dimension can often be problematic in many situations – typically both the computational and sample complexity of algorithms will grow with the embedding dimension of the data. Furthermore, excessive dimensionality can also often lead to overfitting for many supervised learning techniques, wherein the number of free parameters in a model increases with the dimension and the resulting fitted model describes the random noise or error better than the underlying relationships: this is a case of excessive *model complexity* which grows with dimension. Together, these high dimensional phenomena constitute the so-called "curse of dimensionality."

To allay this problem, one approach is to try to reduce the dimension of the data whilst preserving its descriptive content. There are a multitude of such methods – we will focus on linear methods such as PCA, ICA and random projection, and pass over non-linear methods (see, for example, the rather comprehensive book [94]). A related approach, especially in the supervised learning case, is "feature selection" where one tries to select the subset of features which is most informative for subsequent steps. One can view feature selection as linear dimension reduction where the linear map has columns given by some subset of the canonical basis vectors $\{e_1, \ldots, e_n\}$.

In learning theory, this type of feature selection problem was formalised first by A. Blum

[29]. In this problem, one is given points from some distribution over $\{0,1\}^n$, labelled by a Boolean function that depends only on k of the n coordinates. The goal is to learn the relevant k coordinates and the labelling function. Naive enumeration of k subsets of the coordinates leads to an algorithm of complexity roughly n^k. Mossel et al [103] gave an algorithm of complexity roughly $O(n^{0.7k})$ assuming the uniform distribution over $\{0,1\}^n$. Another related problem is that of learning the intersection of k halfspaces in $\mathbb{R}^n$ [27, 28] ($k = 1$ is the classic problem of learning a halfspace). Although the complexity of both problems is far from settled, there has been much progress in recent years for special cases, as we discuss in this section.

One way to synthesis these diverse linear dimension reduction ideas is to observe that feature selection assumes that we know the correct basis for the data, and that it suffices to select columns from the data. Instead, sometimes we often do not know a good basis for the data, and thus it is incumbent upon us to simultaneously select a good basis and a good subset of columns in this basis. To formalise this, we will introduce the following sequence of problems which collectively capture the idea of a good subspace that contains the essential descriptive content of a labelling function.

Let us assume that the data points are drawn from some distribution F in $\mathbb{R}^n$ that can be factored into a product of two independent marginal distributions F_V and F_W on unknown orthogonal subspaces V and $W = V^\perp$, i.e., $F = F_V F_W$. We call such an F *factorizable*. Thus, a random point in F is generated by first picking its coordinates in V according to F_V and then independently picking coordinates in W according to F_W. The corresponding problem is the following.

Problem 5 (Factoring distributions)**.** *Given (unlabelled) samples from a factorizable distribution $F = F_V F_W$ over $\mathbb{R}^n$ (with V and W unknown), recover a factorization of F.*

If F in fact factorizes further into a product of more distributions, or even a full product distribution of one-dimensional component distributions as in ICA, an algorithm for the above problem can be applied recursively to find the full factorization.

The factoring problem above has direct applications to learning in high dimension. Let

π_V denote projection to a subspace V. We consider labelling functions $\ell : \mathbb{R}^n \to \{0,1\}$ of the form $\ell(x) = \ell(\pi_V(x))$. We are given points according to some distribution F over $\mathbb{R}^n$ along with their labels $\ell(x) = \ell(\pi_V(x))$ for some unknown subspace V of dimension k (the 'relevant' subspace), and wish to learn the unknown concept ℓ, i.e., find a function that agrees with ℓ on most of F. We call this the problem of learning a *k-subspace junta*. We further assume that F is factorizable as $F = F_V F_W$, with $W = V^\perp$ (the 'irrelevant' subspace). The justification for this factorizability assumption is that coordinates in the W subspace are not relevant to the labelling function and can be considered to be noisy attributes. The full statement of our learning problem is as follows:

Problem 6 (Learning a k-subspace junta)**.** *For $\epsilon, \delta > 0$, given samples drawn from a factorizable distribution $F = F_V F_W$, and labelled by a $\ell = f \circ \pi_V$, find a* 0-1 *function f such that with probability at least $1 - \delta$,*

$$\Pr_F(\ell(x) \neq f(x)) \leq \epsilon.$$

For special cases of Problem 6, previous authors have applied standard low-dimensional representation techniques, low-degree polynomials, random projection and PCA to identify V under strong distributional assumptions [20, 85, 27, 126]. The strongest result in this line achieves a fixed polynomial dependence on n by applying PCA to learn convex concepts over Gaussian input distributions [125]. Unfortunately, standard PCA does not work for other distributions or more general concept classes, in part because PCA does not provide useful information when the covariance matrices of the positive and negative samples are equal. In fact, the problem appears to be quite hard with no assumptions on the input distribution, even for small values of k, e.g., a single halfspace can be PAC-learned via linear programming, but learning an intersection of two halfspaces (a 2-subspace junta) in polynomial time is an open problem.

The strictest variant of this problem is the case when the the distribution over W, F_W, is a Gaussian distribution (any Gaussian distribution in fact, we do not know the mean or covariance *a priori*). The idea in this case is that the variables in W are truly irrelevant, and are simply measurements of some unrelated phenomena which induce Gaussian

measurement error. We call this the Gaussian noise model, for the obvious reason:

Problem 7 (Gaussian noise model)**.** *For $\epsilon, \delta > 0$, given samples drawn from a factorizable distribution $F = F_V F_W$, where F_W is any Gaussian distribution independent of F_V, and labelled by a $\ell = \ell \circ \pi_V$, find a* 0-1 *function f such that with probability at least $1 - \delta$,*

$$\Pr_F(\ell(x) \neq f(x)) \leq \epsilon.$$

2.5 The power and limitations of PCA

We turn now to one of the oldest, best known and most versatile of unsupervised learning methods: PCA. This will also serve as a springboard in later chapters for the development of higher order generalisations to tensors, which will form the crux of our algorithmic approach to machine learning.

Principal Component Analysis [111] is often an "unreasonably effective" heuristic in practice, and some of its effectiveness can be explained rigorously as well (see, e.g., [82]). It consists of computing the eigenvectors of the empirical covariance matrix formed from the data; the eigenvectors turn out to be directions that locally maximize second moments. More formally, suppose our data is drawn according to some distribution F, then denote $\mu = \mathbb{E}_{x \sim F}(x)$ and $\Sigma = \mathbb{E}_{x \sim F}\left((x - \mu)(x - \mu)^T\right)$ (henceforth, we shall drop the explicit $x \sim F$). Then the top principal component v_1 of F is given by the optimisation problem:

$$v_1 = \max_{v \in \mathbb{R}^n : \|v\| = 1} v^T \Sigma v$$

One can view this as a second moment optimisation as follows (here for the sake of clarity, we shall assume that $\mu = 0$):

$$\max_{v \in \mathbb{R}^n : \|v\| = 1} \mathbb{E}\left((x^T v)^2\right) \tag{1}$$

It is clear then that since Σ is a symmetric matrix, then v is in fact the top eigenvector. We can proceed similarly for all remaining eigenvalues by taking orthogonal projecions. Roughly speaking, the top k (of n) principal components give the orthogonal directions of greatest variance, and in some sense, these are the most "interesting" directions in data. Projection onto the top k principal components or eigenvectors is a way of obtaining a

reduction of dimension from n to k. One important note here is that one can compute all the eigenvalues and eigenvectors of such matrices very efficiently, and thus algorithms which use PCA can typically easily be shown to run in polynomial time.

We will give a brief survey of the best PCA based results for the problems detailed above – with our focus being provably good algorithms. The monograph [82] details a number of applications where one can obtain provably good guarantees on performance.

One illustration of the dimension reduction idea is the algorithm of [127] for the case when all the Gaussians are spherical. Roughly speaking, their algorithm projects the data onto the top k principal components – one can show that this subspace in fact spans the mean vectors $\{\mu_1, \ldots, \mu_k\}$. By projecting to this subspace, one can obtain better separations between the individual pairs of Gaussians, and thus the main result is as follows:

Theorem 2.5.1. *Let F be a Gaussian mixture model as in Problem 3 where:*

1. $\Sigma_i = \sigma_i^2 I_n$ *for each* $i \in [m]$, *and*
2. $\|\mu_i - \mu_j\| \geq Ck^{1/4} \log(n/w_{\min})^{1/4} \max \sigma_i$,

where C is some absolute constant. Then, one can recover all the parameters of the model $\{\mu_i, w_i, \sigma_i\}$ in time polynomial in $k^{O(k)}$ and n

The point here is that by applying PCA-based dimension reduction, the separation in (2) above is in terms of a polynomial of k, and not in n.

For the planted clique problem we have following upper bound which uses SVD [6, 99, 43] of the adjacency matrix of the graph:

Theorem 2.5.2. *Let G be as in Problem 4. Then one can find cliques of size $k = \Omega(\sqrt{n})$.*

This is essentially the best known result for this problem, and proceeds by a simple SVD of adjacency matrix of G followed by a clean-up phase.

The following ICA-type example illustrates the power and limitations of PCA: given random independent points from a rotated cuboid in $\mathbb{R}^n$ with distinct axis lengths, PCA will identify the axes of the cuboid and their lengths as the eigenvectors and eigenvalues of the covariance matrix. However, if instead of a rotation, points came from a general

linear transformation of a cuboid, then PCA does not work. This example is a special case of the ICA problem where all the s_i are given by uniform distributions over $[0, 1]$. Although PCA fails, one can use it to first apply a transformation (to a sample) that makes the distribution isotropic, i.e., effectively making the distribution a rotation of a cube. At this point, eigenvectors give no further information. This in fact is a key limitation of PCA: because we are computing the spectral decomposition of a second moment matrix, we essentially have no access to the higher order moments and finer correlations in the data.

But as observed in the signal processing literature ([91, 44] and the surveys [47, 76]), directions that locally maximize fourth moments reveal the axes of the cube, and undoing the isotropic transformation yields the axes of the original cuboid. Thus in contrast to Equation 1 where the second moment is maximized, we can instead maximize fourth moments:

$$\max_{v \in \mathbb{R}^n : \|v\|=1} \mathbb{E}\left((x^T v)^4\right)$$

Using this basic idea, Frieze et al. [62] and subsequent papers give provably efficient algorithms assuming that the linear transformation A is full-dimensional and the components of the product distribution differ from one-dimensional Gaussians in their fourth moment. To handle this and similar hurdles, higher moment extensions of PCA have been developed in the literature e.g., [9, 74, 104, 11, 8, 73, 129] and shown to be provably effective for a wider range of unsupervised learning problems, including special cases of ICA, Gaussian mixture models, learning latent topic models etc.

Thus, a general limitation to PCA is that it considers only the second moments, and one needs generalisations of PCA to consider higher order information about distributions. There are many ways to do this – one method is simply to reweight the data to interlace some of the higher order information with the second moment [34] (we explore a similar idea in Section 5.3). A different idea is to develop a theory of spectral tensor decompositions in analogy with the spectral theorem for matrices. This is the focus of the next chapter. Note, that these limitations of PCA are not restricted to the ICA example: for planted clique, there are tensor based algorithms which provide stronger upper bounds [63, 32] conditional

on being able to optimize a particular type of tensor; for Gaussian mixtures, one can in fact do quite a bit better using tensor techniques as well (see [9]).

2.6 Contributions of this thesis

In this thesis, we tackle all of the unsupervised learning problems above with positive results. We give an algorithm for fully determined ICA which can be made extremely efficient (both in theory and practice) in terms of sample complexity in Section 5.3, we give an alternative algorithm for resolving mixtures of spherical Gaussians in Section 5.5, and we extend our techniques to give the first provably good underdetermined ICA algorithm in Section 5.6. For the dimensionality reduction or relevant feature model that we defined above, we develop a general algorithmic framework for solving this problem in Section 5.2 and solve a number of actual learning problems that do not appear susceptible to known methods in Section 5.2.5.

What makes these advances possible are the tensor decompositions that we develop in Chaper 4. Therein, we give two provably good, polynomial time, robust tensor decompositions that recover some of the properties of PCA for matrices. The first decomposition recovers weak block-like structures in tensors that express the effect of a tensor as a sum of the effects over two subspaces. This turns out to be a very natural notion and generalises the rank 1 type decompositions that we noted are NP-hard. The second decomposition tackles this NP-hardness by the horns, and we give a natural structural property for rank 1 decompositions that guarantees that we can find it efficiently. The major achievement in the latter case is twofold – we are able to give a decomposition which is able to decompose tensors with more rank 1 components than the ambient dimension n, and we are able to characterise the performance of this algorithm in terms of a single parameter of the rank decomposition. Both these decompositions are put to use solving our unsupervised learning problems.

CHAPTER III

MATHEMATICAL PRELIMINARIES

For positive integer n, the set $\{1, \ldots, n\}$ is denoted by $[n]$. The set of positive even numbers is denoted by $2\mathbb{N}$.

3.1 Probability

Fact 3.1.1. *For a real-valued random variable x and for any $0 < p \leq q$ we have*

$$\mathbb{E}\left(|x|^p\right)^{1/p} \leq \mathbb{E}\left(|x|^q\right)^{1/q},$$

$$\mathbb{E}\left(|x|^p\right)\mathbb{E}\left(|x|^q\right) \leq \mathbb{E}\left(|x|^{p+q}\right).$$

Proof. Hölder's inequality implies that for $0 \leq p \leq q$ we have

$$\mathbb{E}\left(|x|^p\right)^{1/p} \leq \mathbb{E}\left(|x|^q\right)^{1/q},$$

and hence

$$\mathbb{E}\left(|x|^p\right)\mathbb{E}\left(|x|^q\right) \leq \mathbb{E}\left(|x|^{p+q}\right)^{p/(p+q)}\mathbb{E}\left(|x|^{p+q}\right)^{q/(p+q)} = \mathbb{E}\left(|x|^{p+q}\right).$$

□

For a random variable $x \in \mathbb{R}^n$ and $u \in \mathbb{R}^n$, its *characteristic function* $\phi : \mathbb{R} \to \mathbb{C}$ is defined by $\phi_x(u) = \mathbb{E}_x\left(e^{iu^T x}\right)$. Unlike the moment generating function, the characteristic function is well-defined even for random variables without all moments finite. The *second characteristic function* of x is defined by $\psi_x(u) := \log \phi_x(u)$, where we take that branch of the complex logarithm that makes $\psi(0) = 0$. In addition to random variable x above we will also consider random variable $s \in \mathbb{R}^m$ related to x via $x = As$ for $A \in \mathbb{R}^{n\times m}$ and the functions associated with it: the characteristic function $\phi_s(t) = \mathbb{E}_s\left(e^{it^T s}\right)$ and the second characteristic function $\psi_s(t) = \log \phi_s(t)$.

Let $\mu_j := \mathbb{E}\left(x^j\right)$. *Cumulants* of x are polynomials in the moments of x which we now define. For $j \geq 1$, the jth cumulant is denoted $\mathrm{cum}_j(x)$. Some examples: $\mathrm{cum}_1(x) =$

$\mu_1, \text{cum}_2(x) = \mu_2 - \mu_1^2, \text{cum}_3(x) = \mu_3 - 3\mu_2\mu_1 + 2\mu_1^3$. As can be seen from these examples the first two cumulants are the same as the expectation and the variance, resp. Cumulants have the property that for two independent r.v.s x, y we have $\text{cum}_j(x+y) = \text{cum}_j(x) + \text{cum}_j(y)$ (assuming that the first j moments exist for both x and y). The first two cumulants of the standard Gaussian distribution have value 0 and 1, and all subsequent cumulants have value 0. Since ICA is not possible if all the independent component distributions are Gaussians, we need some measure of distance from the Gaussians of the component distributions. A convenient measure turns out to be the distance from 0 (i.e. the absolute value) of the third or higher cumulants. If all the moments of x exist, then the second characteristic function admits a Taylor expansion in terms of cumulants

$$\psi_x(u) = \sum_{j \geq 1} \text{cum}_j(x) \frac{(iu)^j}{j!}.$$

This can also be used to define cumulants of all orders.

Denote by $N(\mu, \Sigma)$ the Gaussian distribution (implicitly over $\mathbb{R}^n$ with mean vector $\mu \in \mathbb{R}^n$ and covariance matrix $\Sigma \in \mathbb{R}^{n \times n}$. The following is a standard tailbound:

Claim 3.1.2. *Let $u \in \mathbb{R}$ be sampled according to $N(0, \sigma^2)$. Then for $\tau > 0$ we have*

$$\mathsf{Pr}\left(|u| > \tau\right) \leq \sqrt{\frac{2}{\pi}} \frac{\sigma^2}{\tau} e^{-\frac{\tau^2}{2\sigma^2}}$$

Proof. Follows from the well-known fact: $\frac{1}{\sqrt{2\pi}} \int_a^\infty e^{-z^2/2} dz \leq \frac{1}{\sqrt{2\pi}} \cdot \frac{1}{a} \cdot e^{-a^2/2}$, for $a > 0$ $\square$

We will now quote some results on polynomials of random variables, and also prove a polynomial anti-concentration inequality.

Lemma 3.1.3 (Schwartz-Zippel[116])**.** *Let $P \in F[x_1, \ldots, x_n]$ be a nonzero polynomial of degree $dn \geq 0$ over field F. Let S be a finite subset of F and let $r_1, \ldots, r_n$ be selected randomly from S. Then:*

$$\mathsf{Pr}\left(P(r_1, \ldots, r_n) = 0\right) \leq \frac{d}{|S|}.$$

Here is a robust version developed using the Carbery-Wright inequality [35]:

Lemma 3.1.4 (Robust Schwartz-Zippel)**.** *Let p be a degree m polynomial over n variables and K a convex body in $\mathbb{R}^n$. If there exists $x \in K$ such that $|p(x)| > \epsilon(2cn)^m$, then for l random points s_i, $\mathsf{Pr}\left(\forall s_i : |p(s_i)| \leq \epsilon\right) \leq 2^{-l}$.*

Proof of Lemma 3.1.4. Let μ denote the uniform measure over K, by Corollary 2 of Carbery and Wright [35]:

$$\max_{x \in K} |p(x)|^{1/m} \epsilon^{-1/m} \mu(\{x \in K : |p(x)| \leq \epsilon\}) \leq cn$$

Consider our l samples – there are two possibilities:

1. $\mu(\{x \in K : |p(x)| \leq \epsilon\}) \geq 1/2$. In this case, we have $|p(x)| \leq \epsilon(2cn)$ from the bound above.

2. $\mu(\{x \in K : |p(x)| \leq \epsilon\}) \leq 1/2$. Then, $\mathsf{Pr}\left(\forall i\, |p(x_i)| \leq \epsilon\right) \leq 1/2^l$.

□

Of course this probability can be amplified by repeating the test (or simply taking l larger).

We will also an anti-concentration inequality for univariate polynomials under a Gaussian measure. While this inequality appears very similar to the inequality of Carbery–Wright [35] (cf. [102], Corollary 3.23), we are not able to derive our inequlity from it. The hypothesis of our inequality is weaker in that it only requires the polynomial to be monic instead of requiring the polynomial to have unit variance as required by Carbery–Wright; on the other hand it applies only to univariate polynomials.

Theorem 3.1.5 (Anti-concentration of a polynomial in Gaussian space)**.** *Let $p(x)$ be a degree d monic polynomial over $\mathbb{R}$. Let $x \sim N(0, \sigma^2)$, then for any $t \in \mathbb{R}$ we have*

$$\mathsf{Pr}\left(|p(x) - t| \leq \epsilon\right) \leq \frac{4d\epsilon^{1/d}}{\sigma\sqrt{2\pi}}.$$

For most of the proof we will work with the Lebesgue measure; the proof for the Gaussian measure will follow immediately. Our starting point is the following lemma which can be derived from the properties of Chebyshev polynomials ([31], Sec 2.1, Exercise 7); we include

a proof for completeness. For interval $[a, b]$, define the supremum norm on real-valued functions f defined on $[a, b]$ by

$$\|f(x)\|_{[a,b]} := \left\|f(x)\chi_{[a,b]}\right\|_\infty = \sup_{x\in[a,b]} |f(x)| .$$

Then we have

Lemma 3.1.6. *The unique degree d monic polynomial minimising $\|p(x)\|_{[a,b]}$ is given by*

$$p(x) = 2\left(\frac{b-a}{4}\right)^d T_d\left(\frac{2x-a-b}{b-a}\right), \tag{2}$$

where T_d is the d^{th} Chebyshev polynomial.

Proof. We already know (see [31]) that for the interval $[-1, 1]$ the unique monic polynomial of degree d which minimizes $\|p(x)\|_{[-1,1]}$ is given by $2^{1-d}T_d(x)$. Map the interval $[a, b]$ to $[-1, 1]$ using the affine map $f(x) = (2x - a - b)/(b - a)$ which satisfies $f(a) = -1$ and $f(b) = 1$. Then $((b-a)/2)^d 2^{1-d}T_d(x) = 2((b-a)/4)^d T_d(x)$ is the unique monic polynomial minimizing $\|\cdot\|_{[a,b]}$. For if it were not, we could use such a polynomial to construct another monic polynomial (by reversing the above transformation) which contradicts the fact that Chebyshev polynomials are the unique monic minimizers of $\|\cdot\|_{[-1,1]}$. □

From this we have the following lemma.

Lemma 3.1.7. *Let $p(x)$ be a degree d monic polynomial over $\mathbb{R}$. Fix $\epsilon > 0$, then for every x, there exists an x' where $|x - x'| \le \epsilon$ and $|p(x) - p(x')| \ge 2(\epsilon/2)^d$.*

Proof. We will translate the polynomial p to obtain the polynomial $q(y)$ as follows:

$$q(y) = p(y + x) - p(x).$$

Observe that $q(y)$ is a monic polynomial and $q(0) = 0$. Now suppose that for all points $x' \in [x - \epsilon, x + \epsilon]$, we have $|p(x) - p(x')| < (\epsilon/2)^d$, then for all $y \in [-\epsilon, \epsilon]$, we must have $|q(y)| < 2(\epsilon/2)^d$.

But, from the previous lemma, we know that for the interval $[-\epsilon, \epsilon]$, the minimum L^∞-norm on the interval for any monic polynomial is attained by $r(y) = 2(\epsilon/2)^d T_d(y/\epsilon)$. The value of this minimum is $2(\epsilon/2)^d$. □

We can use the above lemma to given an upper bound on the measure of the set where a polynomial stays within a constant sized band:

Lemma 3.1.8. *Let $p(x)$ be a degree d monic polynomial. Then for any interval $[a, b]$ where $b - a = \epsilon$ we have*

$$\mu(x \in \mathbb{R} : p(x) \in [a, b]) \leq 4d \left(\frac{\epsilon}{2}\right)^{1/d},$$

where μ denotes the usual Lebesgue measure over $\mathbb{R}$.

Proof. Since p is a continuous function so, $p^{-1}([a, b]) = \cup_i I_i$ where I_i are disjoint closed intervals. There are at most d such intervals: every time $p(x)$ exits and re-enters the interval $[a, b]$ there must be a change of sign in the derivative $p'(x)$ at some point in between. Since $p'(x)$ is a degree $d - 1$ polynomial, there can only be $d - 1$ changes of sign.

Next, suppose that $|I_i| > 4(\epsilon/2)^{1/d}$, then there exists an interval $[x - 2(\epsilon'/2)^{1/d}, x + 2(\epsilon'/2)^{1/d}] \subseteq I_i$, where $\epsilon' > \epsilon$. Then, by applying Lemma 3.1.7, there exists a point x' such that $|x - x'| \leq 2(\epsilon'/2)^{1/d}$ but

$$\begin{aligned} \left|p(x) - p(x')\right| &\geq 2 \left[\frac{1}{2} \cdot 2 \left(\frac{\epsilon'}{2}\right)^{1/d}\right]^d \\ &\geq \epsilon' > \epsilon. \end{aligned}$$

This implies that $x' \notin [a, b]$, which is a contradiction. Hence we must have $|I_i| \leq 4(\epsilon/2)^{1/d}$ and

$$\sum_i |I_i| \leq d \max_i |I_i| \leq 4d(\epsilon/2)^{1/d},$$

as required. □

We can now give the proof for Theorem 3.1.5:

Proof of Theorem 3.1.5. We know that the Lebesgue measure of the set for which $p(x) \in [t - \epsilon, t + \epsilon]$ is given by Lemma 3.1.8. Then multiplying by the maximum density of a Gaussian $1/\sigma\sqrt{2\pi}$ gives us the desired bound. □

In addition to studying all the gaps, we can also study the largest gap between successive eigenvalues of the matrix $D^2 \log(\phi(u))$: instead of requiring that all eigenvalues are well-spaced, we simply require that one adjacent pair of eigenvalues is well-spaced. To this end, for a set of numbers $x_1 \leq \cdots \leq x_n$, define the maximum gap function as:

$$\text{maxgap}\,(x_1, \ldots, x_n) = \max_{i \in [n]} \min_{j \in [n]: x_j \geq x_i} x_j - x_i$$

Thus, we can think of the maxgap function as simply returning the largest gap between two successive elements (in sorted order).

Theorem 3.1.9. *Let $\{p_1(x), \ldots, p_n(x)\}$ be a set of n quadratic polynomials of the form $p_i(x) = a_i x^2$ where $a_i > 0$ for all i. Let $\{z_1, \ldots, z_n\}$ be iid standard Gaussians, and sort $p_i(z_i)$ in ascending order to obtain $p_{i_k}(z_{i_k})$, then:*

$$\text{maxgap}\,(p_1(z_1), \ldots, p_n(z_n)) \geq \frac{\log(2)}{10 \log(n)} \min_i a_i$$

for some absolute constant c, with probability at least $1/2000 \log(n)^2$.

Proof. The first stage of the proof is to reduce the problem from the random model $\{a_1 z_1^2, \ldots, a_n z_n^2\}$ to a mixture model which will more easily allow us to analyse the maximum gaps. To this end, let f_i denote the distribution of $p_i(z_i)$, then consider the following mixture model $F = \frac{1}{n} \sum_{i=1}^n f_i$. One can think of the simulation of a sample $x \sim F$ as a two-stage process. First, we pick an $i \in [n]$ uniformly at random (this gives a corresponding a_i), and then we pick $z \sim N(0, 1)$ independently. The product $a_i z^2$ then has distribution given by F.

Thus, let us pick $m = 10n \log(n)$ samples in this fashion. First we pick m times independently, uniformly at random from $[i]$ (with replacement) to obtain the set $Y = \{y_1, \ldots, y_m\}$. Next we pick m independent standard Gaussian random variables $\{z_1, \ldots, z_m\}$, and finally compute component-wise products $\{y_1 z_1^2, \ldots, y_m z_m^2\}$. We begin with some concentration of measure facts:

Claim 3.1.10. *1.* $\mathsf{Pr}\,(\forall i, a_i \in Y) \geq 1 - \frac{1}{n^9}$

2. $\mathsf{Pr}\,(\exists i, a_i$ *appears more than* $40 \log(n)$ *times in* $Y)$

Proof. The proof of (1) can be found in [105]. The proof of (2) follows from the trivial union bound over all i, and the following standard form of the Chernoff bound for i.i.d. Bernoulli $\{0,1\}$ random variables with bias p

$$\mathsf{Pr}\left(\frac{1}{m}\sum_{i=1}^{m} X_i \geq (1+\delta)pm\right) \leq \exp\left(-\frac{\delta^2 pm}{3}\right) \tag{3}$$

Thus, if we sum the indicator random variables $\chi_{y_i=a_1}$ over all $i \in [m]$, then we take $pm = 10\log(n)$ and $\delta = 3$ which yields a probability bound of $1/n^3$. Now union bounding over all a_j yields the desired answer. □

For the rest of this proof, we shall simply exclude these two events, for a loss of say $2/n^2$ probability. We shall draw a subsample of size n from the set $\{y_1 z_1^2, \ldots, y_m z_m^2\}$ to form the set S. To do so: we bucket the $\{y_i\}$ according to which a_j was picked, and then from each bucket, we pick a single representative uniformly at random . From the claim above, we know that each bucket has at least one element, and at most $40\log(n)$ elements in it. Finally, the set W is simply the variable $y_i z_i^2$ associated with the representative we picked uniformly at random. A trivial observation is that W is distributed exactly as the $\{p_1(z_1), \ldots, p_n(z_n)\}$ in the statement of this theorem. In fact, each a_i shows up exactly once in W, and is multiplied by z^2 for $z \sim N(0,1)$ – all random variables are independent.

Observe that if we pick $\max_i y_i z_i^2$ and $\min_i y_i z_i^2$ are picked for the set W, then it is clear that $\text{maxgap}\,(W) \geq \text{maxgap}\,(y_1, \ldots, y_m)$. This occurs with probability at least $1/1600\log(n)^2$ since no bucket is of size greater than $40\log(n)$. Thus, it suffices for us to analyse $\text{maxgap}\left(y_1 z_1^2, \ldots, y_m z_m^2\right)$. Note that this random variable is independent of what y_j are picked for W, thus we have given a reduction from our original random variable model $\{a_1 z_1^2, \ldots, a_m z_m^2\}$ to (slightly) more samples from a mixture model F.

To lower bound the maximum gap, observe that the density $F(x)$ is continuous and unimodal, taking its maximum at $x = 0$ and decays to 0 as $x \to \infty$. Thus for some t_0 and t_1, we must have that:

$$\mathsf{Pr}\,(x \geq t_0) = \sqrt{\frac{2}{\pi}}\frac{1}{n}, \qquad \mathsf{Pr}\,(x \geq t_1) = \sqrt{\frac{2}{\pi}}\frac{1}{2\sqrt{2}n\log(2n)}$$

In particular, let us expand out the probabilities explicitly:

$$\begin{aligned}\mathsf{Pr}\left(x \geq t_0\right) &= \frac{1}{n}\sum_{i=1}^{n}\mathsf{Pr}\left(a_i z_i^2 \geq t_0\right) \\ &= \frac{2}{n}\sum_{i=1}^{n}\mathsf{Pr}\left(z_i \geq \sqrt{\frac{t_0}{a_i}}\right)\end{aligned}$$

Note that applying the usual Gaussian tail bound, we have:

$$\mathsf{Pr}\left(x \geq t_0\right) \leq \frac{2}{n\sqrt{2\pi}}\sum_{i=1}^{n}\sqrt{\frac{a_i}{t_0}}\exp\left(-\frac{t_0}{2a_i}\right)$$

In particular, combining this with the definition of t_0, this implies that there exists some i such that:

$$\sqrt{\frac{a_i}{t_0}}\exp\left(-\frac{t_0}{2a_i}\right) \geq \frac{1}{n}$$

In particular, for this term, we certainly must have $t_0 \geq a_i$ since the exponent in the exponential is negative. This further implies that we had better have $-t_0/2a_i \geq -\log(n)$ or $t_0 \leq 2a_i \log(n)$. On the other side, using the same reasoning, we must have that for some i:

$$\mathsf{Pr}\left(x \geq t_1\right) \geq \frac{2}{n\sqrt{2\pi}}\sum_{i=1}^{n}\left[\left(\frac{a_i}{t_1}\right)^{1/2} - \left(\frac{a_i}{t_1}\right)^{3/2}\right]\exp\left(-t_1/2a_i\right) \leq \frac{1}{2n}$$

Then there must be one term such that:

$$\left[\left(\frac{a_i}{t_1}\right)^{1/2} - \left(\frac{a_i}{t_1}\right)^{3/2}\right]\exp\left(-\frac{t_1}{2a_i}\right) \leq \frac{1}{2\sqrt{2}n\log(n)}$$

In particular, this implies that $t_1 \geq 2a_i \log(2n)$. Thus, we can bound the probability of the interval $[t_0, t_1]$ as follows:

$$\mathsf{Pr}\left(x \in [t_0, t_1]\right) \leq \sqrt{\frac{1}{\pi}}\frac{1}{n}$$

Thus by applying the Chernoff bound in Equation 3, we obtain that with probability at least $1 - 1/n^{5/6}$, there are at most $20/\sqrt{\pi}\log(n)$ points in the interval. Observe also that we can use the lower bound for $\mathsf{Pr}\left(x \geq t_1\right)$ and the same concentration of measure to show that there are at least $2/\sqrt{\pi}\log(n)$ points beyong t_1.

. Note that the interval is of at size at least $2\log(2)\min a_i$, thus by an averaging argument, there exists an adjacent pair which are at least $\log(2)/10\log(n)$ apart as required.

To compute the failure probability, we note that $1/1600\log(n)^2$ is far smaller than the $1/n^c$ terms elsewhere in the calculation, so we can bound the failure probability by $1/2000\log(n)^2$. □

Note that of course we can simply repeat the experiment $O(\log(n)^3)$ times to obtain a high probability statement here.

3.2 Linear algebra

For a vector $\mu = (\mu_1, \ldots, \mu_n)$, let $\mathrm{diag}(\mu)$ and $\mathrm{diag}(\mu_j)$, where j is an index variable, denote the $n \times n$ diagonal matrix with the diagonal entries given by the components of μ. The singular values of an $m \times n$ matrix will always be ordered in the decreasing order: $\sigma_1 \geq \sigma_2 \geq \ldots \geq \sigma_{\min(m,n)}$. Our matrices will often have rank m, and thus the non-zero singular values will often, but not always, be $\sigma_1, \ldots, \sigma_m$. The span of the columns vectors of a matrix A will be denoted $\mathsf{colspan}(A)$. The columns of a matrix A are denoted $A_1, A_2, \ldots$. The potentially ambiguous but convenient notation A_i^T means $(A_i)^T$. The condition number of a matrix A is $\kappa(A) := \sigma_{\max}(A)/\sigma_{\min}(A)$, where $\sigma_{\max}(A) := \sigma_1(A)$ and $\sigma_{\min}(A) := \sigma_{\min(m,n)}(A)$.

We state the following easy claim without proof.

Claim 3.2.1. *Let $B \in \mathbb{C}^{p\times m}$ with $p \geq m$ and $\mathsf{colspan}(B) = m$. Let $D \in \mathbb{C}^{m\times m}$ be a diagonal matrix. Then*

$$\sigma_m(BDB^T) \geq \sigma_m(B)^2\sigma_m(D).$$

Claim 3.2.2. *For $E \in \mathbb{C}^{m\times m}$ with $\|E\|_F < 1/2$ we have*

$$(I - E)^{-1} = I + E + R,$$

where $\|R\|_F < m\|E\|_F$.

Proof. For $\|E\|_F < 1/2$ we have

$$(I - E)^{-1} = I + E + E^2 + \ldots.$$

Hence

$$\left\|(I-E)^{-1}-(I+E)\right\|_F \leq \left\|E^2\right\|_F \left\|(I-E)^{-1}\right\|_F < m\left\|E\right\|_F.$$

□

We will use a number of singular value pertubation inequalities. The following is a form of Wedin's Theorem from [123] where notions such as the canonical angles etc. used in the statement below are also explained.

Theorem 3.2.3. *Let $A, E \in \mathbb{C}^{m\times n}$ be complex matrices with $m \geq n$. Let A have singular value decomposition*

$$A = [U_1 U_2 U_3] \begin{pmatrix} \Sigma_1 & 0 \\ 0 & \Sigma_2 \\ 0 & 0 \end{pmatrix} [V_1^* V_2^*]$$

and similarly for $\tilde{A} = A + E$ (with conformal decomposition using $\tilde{U}_1, \tilde{\Sigma}_1$ etc). Suppose there are numbers $\alpha, \beta > 0$ such that

1. $\min \sigma(\tilde{\Sigma}_1) \geq \alpha + \beta$,
2. $\max \sigma(\Sigma_2) \leq \alpha$.

Then

$$\left\|\sin(\Phi)\right\|_2, \left\|\sin(\Theta)\right\|_2 \leq \frac{\|E\|_2}{\beta}$$

where Φ is the(diagonal) matrix of canonical angles between the ranges of U_1 and $\tilde{U}_1$ and Θ denotes the matrix of canonical angles between the ranges of U_2 and $\tilde{U}_2$.

We also require the following form of Weyl's Inequality (see [123]):

Lemma 3.2.4. *Let $A, E \in \mathbb{C}^{m\times n}$, then*

$$|\sigma_i(A+E) - \sigma_i(A)| \leq \sigma_1(E)$$

Combining these two:

Lemma 3.2.5. *Let $A \in \mathbb{C}^{n\times n}$ and suppose that $\sigma_i(A) - \sigma_{i+1}(A) \geq \epsilon$ for all i. Let $E \in \mathbb{C}^{n\times n}$ be a matrix where where $\|E\|_2 \leq \delta$. Denote by v_i the right singular vectors of A and $\hat{v}_i$ the right singular vectors of $A + E$, then:*

$$\|v_i - \hat{v}_i\| \leq \frac{\sqrt{2}\delta}{\epsilon - \delta}$$

Proof. We first write:

$$\|v_i - \hat{v}_i\|^2 = \langle v_i - \hat{v}_i, v_i - \hat{v}_i\rangle = 2(1 - \langle v_i, \hat{v}_i\rangle) = 2(1 - \cos(\theta)) \leq 2(1 - \cos(\theta)^2) = 2\sin(\theta)^2$$

By Weyl's inequality (Lemma 3.2.4), we know that $\left|\sigma(\Sigma_1) - \sigma(\tilde{\Sigma}_2)\right| \geq \epsilon - \delta$. Similarly for the smallest singular value. By Wedin's theorem, we pick the partition Σ_1 to be the top i singular values, with Σ_2 the remaining ones. Thus, taking $\alpha = \sigma_{i+1}(A)$ and $\beta = \epsilon - \delta$, we have

$$|\sin(\theta)| \leq \|\sin(\Phi)\|_2 \leq \frac{\delta}{\epsilon - \delta}$$

as required. □

We now prove a harder inequality for general complex matrices, not simply symmetric, Hermitian or normal matrices. giving a robust version of our algorithms is that the stability of eigenvectors of general matrices is more complicated than for Hermitian or normal matrices where the $\sin(\theta)$ theorem of Davis and Kahan [51] describes the whole situation. Roughly speaking, the difficulty lies in the fact that for a general matrix, the eigenvalue decomposition is given by $A = PDP^{-1}$. Upon adding a perturbation E, it is not clear *a priori* that $A + E$ has a full set of eigenvectors—that is to say, $A + E$ may no longer be diagonalizable. The goal of this section is to establish that for a general matrix with well-spaced eigenvalues, sufficiently small perturbations do not affect the diagonalizability. We use Bauer-Fike theorem via a homotopy argument typically used in proving strong versions of the Gershgorin Circle Theorem [132].

Theorem 3.2.6 (Bauer-Fike [19])**.** *Let $A \in \mathbb{C}^{n\times n}$ be a diagonalizable matrix such that $A = X\mathrm{diag}\,(\lambda_i)\,X^{-1}$. Then for any eigenvalue μ of $A + E \in \mathbb{C}^{n\times n}$ we have*

$$\min_i |\lambda_i(A) - \mu| \leq \kappa(X)\,\|E\|_2\,.$$

Using this, we prove a weak version of Weyl's theorem for diagonalizable matrices whose eigenvalues are well-spaced. We consider this a spectral norm version of the strong Gershgorin Circle theorem (which uses row-wise L^1 norms).

Lemma 3.2.7 (Generalized Weyl inequality)**.** *Let $A \in \mathbb{C}^{n\times n}$ be a diagonalizable matrix such that $A = X\mathrm{diag}\,(\lambda_i)\,X^{-1}$. Let $E \in \mathbb{C}^{n\times n}$ be a matrix such that $|\lambda_i(A) - \lambda_j(A)| \geq 3\kappa(X)\,\|E\|_2$ for all $i \neq j$. Then there exists a permutation $\pi : [n] \to [n]$ such that*

$$\left|\lambda_i(A+E) - \lambda_{\pi(i)}(A)\right| \leq \kappa(X)\,\|E\|_2\,.$$

Proof. Consider the matrix $M(t) = A + tE$ for $t \in [0,1]$. By the Bauer-Fike theorem, every eigenvalue $\hat{\lambda}(t)$ of $M(t)$ is contained in $\mathbb{B}(\lambda_i, t\kappa(X)\,\|E\|_2)$ for some i (for $\lambda \in \mathbb{C}$, $t \in \mathbb{R}$ we use $\mathbb{B}(\lambda, t)$ to denote the ball in $\mathbb{C}$ of radius t with center at λ). In particular, when $t = 0$ we know that $\hat{\lambda}(0) = \lambda_i \in \mathbb{B}(\lambda_i, 0)$.

As we increase t, $\hat{\lambda}(t)$ is a continuous function of t, thus it traces a connected curve in $\mathbb{C}$. Suppose that $\hat{\lambda}(1) \in \mathbb{B}(\lambda_j, \kappa(X)\,\|E\|_2)$ for some $j \neq i$, then for some t^*, we must have $\hat{\lambda}(t^*) \notin \bigcup_j \mathbb{B}(\lambda_i, \kappa(X)\,\|E\|_2)$ as these balls are disjoint. This contradicts the Bauer-Fike theorem. Hence we must have $\hat{\lambda}(1) \in \mathbb{B}(\lambda_i, \kappa(X)\,\|E\|_2)$ as desired. □

The following is a sufficient condition for the diagonalizability of a matrix. The result is well-known (Exercise V.8.1 in [93] for example).

Lemma 3.2.8. *Let $A : V \to V$ be a linear operator over a finite dimensional vector space of dimension n. Suppose that all the eigenvalues of A are distinct, i.e., $\lambda_i \neq \lambda_j$ for all pairs i, j. Then A has n linearly independent eigenvectors.*

We require the following generalisation of the Davis-Kahan $\sin(\theta)$ theorem [51] for general diagonalizable matrices due to Eisenstat and Ipsen [56]:

Theorem 3.2.9 (Generalized $\sin(\theta)$ Theorem)**.** *Let $A, A+E \in \mathbb{C}^{n\times n}$ be diagonalizable matrices. Let $\hat{\gamma}$ be an eigenvalue of $A+E$ with associated eigenvector $\hat{x}$. Let*

$$A = [X_1, X_2] \begin{pmatrix} \Gamma_1 & 0 \\ 0 & \Gamma_2 \end{pmatrix} [X_1, X_2]^{-1},$$

be an eigendecomposition of A. Here Γ_1 *consists of eigenvalues of A closest to* $\hat{\gamma}$*, i.e.* $\|\Gamma_1 - \hat{\gamma}I\|_2 = \min_i |\gamma_i - \hat{\gamma}|$*, with associated matrix of eigenvectors* X_1*. And* Γ_2 *contains the remaining eigenvalues and the associated eigenvectors are in* X_2*. Also let* $[X_1, X_2]^{-1} =: \begin{pmatrix} Z_1^* \\ Z_2^* \end{pmatrix}$.

Then the angle between $\hat{x}$ *and the subspace spanned by the eigenvectors associated with* Γ_1 *is given by*

$$\sin(\theta) \leq \kappa(Z_2) \frac{\|(A - \hat{\gamma}I)\hat{x}\|_2}{\min_i |(\Gamma_2)_{ii} - \hat{\gamma}|}.$$

Moving on, we will now introduce the notation that we shall use for tensors. These are discussed in detail in the review paper [86]. An order d tensor T is an array indexed by d indices each with n values (e.g., when $d = 2$, then T is simply a matrix of size $n \times n$). Thus, it has n^d entries. Tensors considered in this paper are symmetric, i.e. $T_{i_1,\ldots,i_d}$ is invariant under permutations of $i_1, \ldots, i_d$. In the sequel we will generally not explicitly mention that our tensors are symmetric. We also note that symmetry of tensors is not essential for our results but for our application to ICA it suffices to look at only symmetric tensors and the results generalize easily to the general case, but at the cost of additional notaton. A good example of a tensor (which we shall see again later) is the moment tensor: for a random vector $x \in \mathbb{R}^n$ with distribution F, the m^{th} moment tensor M^m is a tensor of order m with n^m entries given by:

$$M^m_{i_1,\ldots,i_m} = \mathbb{E}\left(x_{i_1} \ldots x_{i_m}\right).$$

We can also view a tensor as a degree-d homogeneous form over vectors $u \in \mathbb{R}^n$ defined by $T(u, \ldots, u) = \sum_{i_1,\ldots,i_d} T_{i_1,\ldots,i_d} u_{i_1} \cdot \cdots u_{i_d}$. This is in analogy with matrices, where every matrix A defines a quadratic form, $u^T Au = A(u, u) = \sum_{i,j} A_{i,j} u_i u_j$. The following tells us that for symmetric tensors (ones where $T_{i_1,\ldots,i_r} = T_{i_{\sigma(1)},\ldots,i_{\sigma(r)}}$ for any permutation $\sigma : [r] \to [r]$):

Claim 3.2.10. *If* T *is a symmetric order* r *tensor, then:*

$$\max_{\|v\|=1} T(v, \ldots, v) = \max_{\|v_1\|=1,\ldots,\|v_r\|=1} T(v_1, \ldots, v_r)$$

Proof. Clear by induction on r, starting at the case when $r = 2$ for symmetric matrices M where optimising $u^T Mv$ is the same as optimising $v^T Mv$ □

We use the outer product notation

$$v^{\otimes d} = \underbrace{v \otimes \cdots \otimes v}_{d \text{ copies}},$$

where entrywise we have $[v \otimes \cdots \otimes v]_{j_1,\ldots,j_d} = v_{j_1} \cdots v_{j_d}$. A (symmetric) rank-1 decomposition of a tensor T_μ is defined by

$$T_\mu = \sum_{i=1}^{m} \mu_i A_i^{\otimes d}, \tag{4}$$

where the $\mu_i \in \mathbb{R}$ are nonzero and the $A_i \in \mathbb{R}^n$ are vectors which are not multiples of each other. Such a decomposition always exists for all symmetric tensors with $m < n^d$ (better bounds are known but we won't need them). For example, for a symmetric matrix, by the spectral theorem we have

$$M = \sum_{i=1}^{n} \lambda_i v_i \otimes v_i.$$

We will use the notion of flattening of tensors. Instead of giving a formal definition it's more illuminating to give examples. E.g. for $d = 4$, construct a bijection $\tau : [n^2] \to [n] \times [n]$ as $\tau(k) = (\lfloor k/n \rfloor, k - \lfloor k/n \rfloor)$ and $\tau^{-1}(i, j) = ni + j$. We then define a packing of a matrix $B \in \mathbb{R}^{n \times n}$ to a vector $p \in \mathbb{R}^{n^2}$ by $B_{\tau(k)} = p_k$. For convenience we will say that $B = \tau(p)$ and $p = \tau^{-1}(B)$. We also define a packing of $T \in \mathbb{R}^{n \times n \times n \times n}$ to a matrix $M \in \mathbb{R}^{n^2 \times n^2}$ by $M_{a,b} = T_{\tau(a),\tau(b)}$, for $a, b \in [n^2]$. Note that M is symmetric because T is symmetric with respect to all permutations of indices: $M_{a,b} = T_{\tau(a),\tau(b)} = T_{\tau(b),\tau(a)} = M_{b,a}$. The definition of τ depends on the dimensions and order of the tensor and what it's being flattened into; this will be clear from the context and will not be further elaborated upon. Finally, to simplify the notation, we will employ the Khatri-Rao power of a matrix: $A^{\odot d} := \left[\text{vec}\left(A_1^{\otimes d}\right) | \text{vec}\left(A_2^{\otimes d}\right) | \ldots | \text{vec}\left(A_m^{\otimes d}\right)\right]$, where recall that $\text{vec}\,(T)$ for a tensor T is a flattening of T, i.e. we arrange the entries of T in a single column vector.

3.3 Calculus

For $g : \mathbb{R}^n \to \mathbb{R}$ we will use abbreviation $\partial_{u_i} g(u_1, \ldots, u_n)$ for $\frac{\partial g(u_1,\ldots,u_n)}{\partial u_i}$; when the variables are clear from the context, we will further shorten this to $\partial_i g$. Similarly, $\partial_{i_1,\ldots,i_k} g$ denotes $\partial_{i_1}(\ldots(\partial_{i_k} g)\ldots)$, and for multiset $S = \{i_1, \ldots, i_k\}$, this will also be denoted by $\partial_S g$, which makes sense because $\partial_{i_1,\ldots,i_k} g$ is invariant under reorderings of $i_1, \ldots, i_k$. We will not use any special notation for multisets; what is meant will be clear from the context. However, we typically choose to be coordinate free and simply denote the derivative operator with D. For a multivariate function f, Df defines a vector field, and Df_u is its value at any point $u \in \mathbb{R}^n$. In general, $D^r f_u$ will denote the tensor of order r derivatives evaluated at the position u.

CHAPTER IV

TENSOR ALGORITHMS

4.1 Introduction

The spectral decomposition for Hermitian matrices (or the SVD) has two essential properties that we would like to recover in a tensor setting: the first is the optimisation formulation – that the top eigenvalue is the unit vector v which maximises $v^T Av$, the second is the rank decomposition that $A = \sum_{i=1}^{n} \lambda_i v_i v_i^T$. These two properties are strongly related in the matrix setting via the spectral theorem. Unfortunately, this is not the case for tensors – [72] gives an example where subtracting a multiple of v which optimises the multilinear form defined by a tensor actually increases the rank of the tensor!

This type of complication highlights the subtle pitfalls in generalising the spectral theorem to tensors: there are in fact many more complications: for the tensor optimisation problem, there are a number of hardness results for the natural program:

$$\max_{v \in \mathbb{R}^n : \|v\|=1} T(v, \ldots, v) = \sum_{i_1, \ldots, i_r} T_{i_1, \ldots, i_r} v_{i_1} \cdots v_{i_r}$$

It is NP-hard to solve this problem [72, 69], and in fact it is hard to approximate: for $\alpha > 16/17$, it is NP-hard to approximate the optimum to better than factor $\alpha^{\lfloor r/4 \rfloor}$ [33], and the best known approximation factor is roughly $n^{r/2}$. The notion of tensor rank also poses a number of problems – the space of tensors of rank r is not even topologically closed, and solving an optimisation function over a non-closed space does not guarantee a maximum or a minimum inside the space. More concretely, even rank 1 approximation of tensors is NP-hard [72]:

$$\min_{v \in \mathbb{R}^n, \lambda \in \mathbb{R}^n} \|T - \lambda v \otimes \cdots \otimes v\|$$

Thus, to overcome the fundamental limitations of matrix methods illustrated in Section 2.5 is no straight-forward affair, and consequently many approaches to a tensor spectral theory have been explored. In general, it seems impossible to capture all the desirable

properties of the spectral theorem, and thus research efforts have typically focused on specific properties that one wishes to maintain for tensors – the variational formulation [96], the characteristic polynomial [113], or the rank decomposition (see the survey [86] regarding this dominant approach). Algorithmically, there have been a number of heuristic optimisation algorithms proposed [87], but none of these are known to have polynomial complexity. In contradistinction are results such as [63, 32] that give algorithms for computational problems conditional on the solution to certain tensor problems. Thus, the literature on tensor algorithms is incredibly diverse, but there is a dearth of compelling theoretical results. Prior to our work [128, 65], the only provably good algorithms were for the very special case of tensor decomposition when the tensor is guaranteed to have the form:

$$T = \sum_{i=1}^{n} \lambda_i v_i \otimes \cdots \otimes v_i \tag{5}$$

where all the $v_i \in \mathbb{R}^n$ are orthonormal and $m = n$. Note that this is incredibly limiting and fails to capture the full generality of tensor decompositions – in fact, m should be allowed to range all the way to $\Omega(n^{r-1})$ before the problem becomes ill-posed (this is essentially what the Kruskal rank [89] theorem says, and is "obvious" by simple degree of freedom counting). Our work dispenses with this $m = n$ requirement and allows us to solve the tensor decomposition problem when $m > n$ and indeed when m is much greater. The algorithmics of the $m = n$ case are studied in [62, 107, 9, 8]. The most interesting algorithm amongst these is a tensor version of power iteration. Briefly: if one wants to compute the top eigenvalue/eigenvector of a matrix A, one can start with a random vector $v_0 \in \mathbb{R}^n$ of norm 1. Then, one can update v by applying A so that:

$$v_{k+1} = \frac{Av_k}{\|Av_k\|}$$

The tensor version of this is similar – one can think of Av_k in the above as $A(v_k, \cdot)$ by treating A as an order 2 tensor. In analogy then, for a tensor T:

$$v_{k+1} = \frac{T(v_k, \ldots, v_k, \cdot)}{\|T(v_k, \ldots, v_k, \cdot)\|}$$

Remarkably, one can prove that this works rather well, but unfortunately, one must (morally) project out the stationary point v at the end of such a process. In n dimensions, one can

of course only project out directions n times before we're in a 0 dimensional space, so this method fails to extend to the general case.

In this chapter, we develop fundamental tensor tools – both of the major results in this chapter are ways of decomposing tensors into simpler atoms. In particular, we focus on the robustness of our algorithms, for typically one has to construct the tensors by a sampling or random process prone to noise, and we shall show that even in the presence of these inaccuracies, that our algorithms will still work and have polynomial complexity.

Our first tensor decomposition in Section 4.3 tackles problems in the form of Equation 5. We make a very natural structural assumption (one that even holds true generically as shown in Section 4.3.4), and then go on to give a robust algorithm for recovering the rank 1 components. Our major achievement here is twofold – we are able to extract more rank 1 components than the ambient dimension (i.e., more than n rank 1 components), and we are able to prove that our algorithm is efficient in terms of a single parameter of the rank decomposition. Unfortunately, computing a rank 1 decomposition like this is NP-hard in general, and in some situations, we have a more general structure where the set of $\{v_i\}$ can be partitioned into two orthogonal subspaces, say V and W, then the tensor T has an additive decomposition over these two subspaces:

$$T(u, \ldots, u) = T_V(\pi_V(u), \ldots, \pi_V(u)) + T_W(\pi_W(u), \ldots, \pi_W(u))$$

where T_V, T_W are simply some unknown tensors of appropriate order. We give a second tensor decomposition to recover T_V and T_W in this situation in Section 4.2.

4.2 Additive subspace tensor decomposition

The following is based on [128].

Suppose that we have a tensor which we shall assume *a priori* has the following form:

$$T = \sum_{i=1}^{m_1} \lambda_i v_i \otimes \cdots \otimes v_i + \sum_{i=1}^{m_2} \mu_i w_i \otimes \cdots \otimes w_i$$

with $v_i \in V$ and $w_i \in W$ where V and W are orthogonal subspaces. Then, when we compute the multilinear form $T(u, \ldots, u)$, we can decompose the result into two parts –

one part depends only on the projection of u to V (i.e., the inner products $\langle u, v_i \rangle$) and the other depends only on the projection of u to W. In particular, we can rewrite T as follows:

$$\begin{aligned} T &= \sum_{i=1}^{m_1} \lambda_i \langle u, v_i \rangle^r + \sum_{i=1}^{m_2} \mu_i \langle u, w_i \rangle^r \\ &= T_V(\pi_V(u), \ldots, \pi_V(u)) + T_W(\pi_W(u), \ldots, \pi_W(u)) \end{aligned} \tag{6}$$

One can think of the tensor decomposition of Equation 5 as simply a version of this where each subspace is simply one-dimensional (i.e, we have n subspaces V_i each one of which is spanned by the vector v_i), and thus for this case, we can give a polynomial-time algorithm for recovering the subspaces. On the other hand, it is not true in general that one can decompose a tensor into a small number of rank 1 components, thus a very natural question is whether one can recover the additive subspace structure of a tensor T as in Equation 6. Our first tensor decomposition answers this in the affirmative.

4.2.1 Structural theorem

Let T denote a symmetric order m tensor with additive subspace structure over the subspace V and its orthogonal complement $W = V^\perp$. Thus, for any vector $u \in \mathbb{R}^n$:

$$T(u, \ldots, u) = T_V(\pi_V(u), \ldots, \pi_V(u)) + T_W(\pi_W(u), \ldots, \pi_W(u))$$

where T_V and T_W are order m symmetric tensors. The tensor T naturally induces a m-homogeneous function $f(u) = T(u, \ldots, u)$. This lemma, and its proof, are the crux of this tensor decomposition – roughly speaking what we're doing is shifting mass between the two subspaces and finding that if we already have more mass on V, then shifting even more pass (because of the high polynomial powers) makes it even more advantageous.

Lemma 4.2.1 (Support)**.** *Let T be an additive subspace tensor of order m, then for a local maximum (local minimum) u^* of $f(u) = T(u, \ldots, u)$ restricted to the unit sphere, where $f(u^*) > 0$ ($f(u^*) < 0$), either $\|u_V^*\| = 1$ or $\|u_W^*\| = 1$.*

Proof. Note first that we can represent $f(u)$ as follows by m-homogeneity of tensors involved:

$$f(u) = \|u_V\|^m T_V(u_V^0, \ldots, u_V^0) + \|u_W\|^m T_W(u_W^0, \ldots, u_W^0).$$

Consider the curve $C = \{s(u_V^*)^0 + t(u_W^*)^0 : s^2 + t^2 = 1, s \geq 0, t \geq 0\}$. The point u^* lies on C: thus if u^* is a local maximum in full space, it had better be a local maximum on C. On the other hand, we will show that there are no local maxima interior to C, whence we must have $\|u_V^*\| = 1$ or $\|u_W^*\| = 1$.

Let us denote $a_v = T_V((u_V^*)^0, \ldots, (u_V^*)^0)$ and $a_w = T_W((u_W^*)^0, \ldots, (u_W^*)^0)$. By the assumption that $f_m(u^*) > 0$, we know that least one of a_v or a_w is positive. Suppose that $s \neq 0$ and $s \neq 1$: we form the associated Lagrangian with positive real multiplier λ:

$$\mathcal{L} = a_v s^m + a_w t^m - \lambda(s^2 + t^2 - 1)$$

At every critical point in the interior of C, we must have $D\mathcal{L} = 0$:

$$\begin{pmatrix} m a_v s^{m-1} - 2\lambda s \\ m a_w t^{m-1} - 2\lambda t \end{pmatrix} = 0$$

If we consider only the interior critical points where $s, t > 0$, then both $a_v > 0$ and $a_w > 0$ (otherwise we would have $\lambda > 0$ and $\lambda \leq 0$). There is only one solution:

$$s = \frac{a_w^{1/(m-2)}}{\sqrt{a_v^{2/(m-2)} + a_w^{2/(m-2)}}} \qquad t = \frac{a_v^{1/(m-2)}}{\sqrt{a_v^{2/(m-2)} + a_w^{2/(m-2)}}} \qquad \lambda = (m/2)(a_v s^m + a_w t^m)$$

If we now consider the Hessian on the tangent plane orthogonal to the gradient of the constraint (equivalent to considering the bordered Hessian) , we see that it is positive definite for $m > 3$ (when $m = 3$, differentiating $a_v s^3 + a_w(1 - s^2)^{1.5}$ twice at the critical point gives a positive value):

$$D^2\mathcal{L} = \begin{pmatrix} m(m-1)a_v s^{m-2} - 2\lambda & 0 \\ 0 & m(m-1)a_w t^{m-2} - 2\lambda \end{pmatrix} = \frac{m(m-3)a_v a_w}{\left[a_v^{2/(m-2)} + a_w^{2/(m-2)}\right]^{(m-2)/2}} I > 0$$

In particular, there are no local maxima interior to C, that is, $\|u_V^*\| = 1$ or $\|u_V^*\| = 0$. $\square$

The rest of this section is devoted into turning this insight into an algorithm, and then successively hardening it against algorithmic approximation error, and then against input error, thereby obtaining a fully efficient and robust algorithm.

4.2.2 Algorithm

Our two basic algorithms exploit the property that local optimum to $f(u)$ on the unit sphere must lie in either V or W (Lemma 4.2.1). In this section, we assume that the algorithms have access to exact tensors and can compute exact local optima. We provide efficient algorithms (with error analysis) later. This section captures the essential algorithmic ideas – the subsequent chapters are to harden the algorithm to low magnitude input errors.

The basic idea of the algorithm is local search: start with a random direction and evaluate the j'th moment in that direction. If it is nonzero, then we go to a local max or local min (whichever keeps it bounded away from zero) and thus find a vector of interest; if many random unit vectors take on the value zero, then all directions are in fact zero moments due rigidity of polynomials via the Schwartz-Zippel Lemma (Lemma 3.1.3 and we go to the next higher moment. At the end of the algorithm we have a subset of an orthogonal basis consistent with V and W, and the property that all orthogonal directions have Gaussian moments.

Algorithm 1 FindBasis

Input: Tensor T

1: Orthonormal vectors $B \leftarrow \phi$.
2: **while** $|B| < n$ **do**
3: Compute the tensor T_j^B orthogonal to B, so that for any $v \in B^\perp$, $T(v) = T(v_B, \ldots, v_B)$.
4: **if** $f_j(v/\|v\|) \equiv 0$ **then**
5: **return** B
6: **else**
7: **if** $f(v) > 0$ for some v **then**
8: Compute a local maximum u^* to f starting from v.
9: **else**
10: Compute a local minimum u^* to f starting from v.
11: $B \leftarrow B \cup \{u^*\}$.
12: **return** B

For Line 3, let $A : \mathbb{R}^n \to \mathbb{R}^n$ denote the linear map that projects orthogonal to B. Then

$$T(Au, \ldots, Au) = \sum_{i_1,\ldots,i_m} T_{i_1,\ldots,i_m}(Au)_{i_1} \cdots (Au)_{i_m} = \sum_{j_1,\ldots,j_m} \left(\sum_{i_1,\ldots,i_m} T_{i_1,\ldots,i_m} A_{i_1,j_1} \cdots A_{i_m,j_m} \right) u_{j_1} \cdots u_{j_m}$$

The identity check in Line 4 is performed by selecting a random vector x with i.i.d. uniform coordinates from $\{1, \ldots, 2m\}$ and evaluating the polynomial $f(x)$. Repeating $O(\log(n/\delta))$

times gives a $1-\delta$ probability of success.

Theorem 4.2.2 (Find Basis)**.** *Let T be an additive subspace tensor over subspaces $V, W \subset \mathbb{R}^n$. Then, with probability at least $1-\delta$, each vector in the output of* ***FindBasis*** *lies in either V or W.*

Proof of Theorem 4.3.4. From the above comment, at each step, with probability at least $1-\delta/n$ (hence total failure probability δ), we are able to find a point u where $f(u) \neq 0$. In particular, if $f(u) > 0$, then we find a local maximum u^*. By Lemma 4.2.1, u^* is contained entirely within V or W. The analysis is identical when our initial point u satisfies $f(u) < 0$.

Observe that projecting out span (B) preserves the additive structure of T. Hence a local optimum in $B^\perp$ also will lie in either V or W. □

4.2.3 Local search

To compute approximate local optima, we perform gradient ascent, moving in the direction of the gradient. If moving along the gradient does not increase the function value by a certain value, we switch to second-order moves based on the Hessian. We will use the notation that Dg_u is the gradient of g at u and $D^2 g_u$ for the Hessian matrix. The top eigenvalue of a matrix on a subspace orthogonal to a vector can be computed via a coordinate transformation. We shall denote $M = \|T\|_2$. **LocalOpt** terminates in polynomial time when the parameters ϵ_1,

Algorithm 2 LocalOpt

Input: Function g, error parameter ϵ_1,

$u \leftarrow$ uniformly at random over unit sphere.
while $|\langle u, Dg_u\rangle| \leq (1-\epsilon_1)\|Dg_u\|$ or $\lambda_{\max}(D^2 g_u) \geq \epsilon_2$ on $u^\perp$ **do**
  if $|\langle u, Dg_u\rangle| \leq (1-\epsilon_1)\|Dg_u\|$ **then**
    Direction $v \leftarrow \pi_{u^\perp}(Dg_u)$.
    $u \leftarrow u + r_1 v/\|v\|$.
    Renormalize $u \leftarrow u/\|u\|$.
  else if $\lambda_{\max}(D^2 g_u) \geq \epsilon_2$ on $u^\perp$ **then**
    Direction $v \leftarrow$ top eigenvector of $D^2 g_u$ on $u^\perp$.
    $u \leftarrow u - r_2 v/\|v\|$.
    Renormalize $u \leftarrow u/\|u\|$.
return u.

r_1, ϵ_2 and r_2, the thresholds and step sizes for the first-order moves and second-order moves

are chosen appropriately. Note that ϵ_2 varies with the function value, but the remaining parameters are fixed.

Lemma 4.2.3 (Local search termination)**.** *Let $g(u)$ satisfy $g(tu) = t^m g(u)$ for some integer m. Suppose that for our starting point u that $g(u) \geq \eta > 0$. Choose the parameters as follows:*

$$\epsilon_1 \leq \left(\frac{81m(m-1)^2\eta^2}{1048M}\right)^2 \qquad r_1 \leq \frac{\sqrt{\epsilon_1}}{4m^2M}$$
$$\epsilon_2 = \frac{3m(m-1)g(u)}{4} \qquad r_2 \leq \frac{9\eta}{256(m-2)M}$$

where M is an upper bound for g on the unit sphere. Then ***LocalOpt*** *will terminate in at most* $\text{poly}(M, m, 1/\epsilon_1, 1/r_1, 1/r_2, 1/\eta)$ *iterations.*

Proof of Lemma 4.2.3. Consider an iteration of the algorithm where the first derivative condition is unsatisfied, and we make a step of size r_1 in the direction of $v/\|v\|$ (call the step h). The new function value at this point $u + h$ is given by the Taylor series expansion with error (where ζ lies between u and $u + h$):

$$g(u+h) = g(u) + Dg_u \cdot h + \frac{1}{2}h^T(D^2 g_\zeta)h$$

The increase in function value is lower bounded as follows:

$$\begin{aligned} Dg_u \cdot h + \frac{1}{2}h^T D^2 g_\zeta h &\geq r_1 \left\langle Dg_u, \frac{v}{\|v\|} \right\rangle - \frac{1}{2}r_1^2 (v/\|v\|)^T D^2 g_\zeta (v/\|v\|) \\ &\geq r_1\sqrt{\epsilon_1} - \frac{1}{2}r_1^2 m^2 M \\ &\geq r_1\sqrt{\epsilon_1} - \frac{1}{8}r_1\sqrt{\epsilon_1} \\ &\geq \frac{7}{8}r_1\sqrt{\epsilon_1} \end{aligned}$$

Thus, we have lower bounded the increase in the function value. When we rescale $u + h$ back to norm 1, we can apply the m-homogeneity of f to deduce that:

$$g\left(\frac{u+h}{\|u+h\|}\right) = \frac{1}{\|u+h\|^{m/2}} g(u+h)$$

We can compute $\|u+h\| = 1 + r_1^2$ because r_1 is perpendicular to u. Hence:

$$\begin{aligned} g\left(\frac{u+h}{\|u+h\|}\right) &= \frac{1}{(1+r_1^2)^{m/2}} g(u+h) \\ &\geq \left(1 - \frac{m+2}{2} r_1^2\right) g(u+h) \\ &\geq g(u) \left(1 + \frac{7}{8g(u)} r_1 \sqrt{\epsilon_1}\right) \left(1 - \frac{m+2}{2} r_1^2\right) \end{aligned}$$

where we used the estimate:

$$(1+x)^k \geq 1 + (k+1/2)x$$

for $x \leq 2/k^2$. To finish this calculation, we simply substitute our value for r_1 in terms of ϵ_1:

$$\begin{aligned} g\left(\frac{u+h}{\|u+h\|}\right) &\geq g(u) \left(1 + \frac{7}{8g(u)} r_1 \sqrt{\epsilon_1}\right) \left(1 - \frac{1}{8M} r_1 \sqrt{\epsilon_1}\right) \\ &\geq g(u) \left(1 + \frac{5}{8M} r_1 \sqrt{\epsilon_1}\right) \end{aligned}$$

Hence, there are at most at most a polynomial number of iterations of this form. Consider now an iteration where the second derivative condition is unsatisfied (and the first derivative condition must be satisfied). We take the same Taylor series expansion with error term (to one further term), where $h = r_2 v / \|v\|$:

$$g(u+h) = g(u) + Dg_u \cdot h + \frac{1}{2} h^T D^2 g_u h + \frac{1}{6} D^3 g_\zeta(h,h,h)$$

We will show that the contributions from the first and third derivative terms are small, and that the second derivative term dominates. In the first derivative term, note that h is orthogonal to u, and hence the component of Dg_u parallel to h has norm at most $\sqrt{2\epsilon_1 - \epsilon_1^2}\,\|Dg_u\|$. We estimate the other terms as before:

$$\begin{aligned} Dg_u \cdot h + \frac{1}{2} h^T D^2 g_u h + \frac{1}{6} D^3 g_\zeta(h,h,h) &\geq -\sqrt{2\epsilon_1 - \epsilon^2} m M + \frac{1}{2} r_2^2 \epsilon_2 - \frac{m(m-1)(m-2)}{6} r_2^3 M \\ &\geq -\frac{1}{128} r_2^2 \epsilon_2 + \frac{1}{2} r_2^2 \epsilon_2 - \frac{1}{128} r_2^2 \epsilon_2 \\ &\geq \frac{31}{64} r_2^2 \epsilon_2 \end{aligned}$$

Once again, we have to rescale back to norm 1. In this case:

$$\begin{aligned} g\left(\frac{u+h}{\|u+h\|}\right) &\geq g(u)\left(1+\frac{31}{64g(u)}r_2^2\epsilon_2-\frac{m+1}{2}r_2^2\right) \\ &\geq g(u)\left(1+\frac{93}{256}m(m-1)r_2^2-\frac{m+1}{2}r_2^2\right) \\ &\geq g(u)\left(1+\frac{93}{256}r_2^2\right) \end{aligned}$$

The last bound follows because the worst possible lower bound occurs at $m = 3$. Hence, there are only a polynomial number of iterations of this form as well. □

4.2.4 Exact tensor, approximate local optima

We are now ready to extend the analysis of Theorem 4.3.4 to the case when we have access to the exact tensor, but instead of using exact optima, we will use **LocalOpt** with appropriately chosen ϵ_1. On the other hand, using a weaker local optimum algorithm will also give us weaker guarantees on the quality of the output, giving a weaker form of Lemma 4.2.1.

Lemma 4.2.4 (Exact tensor, inexact optima)**.** *Let T be an additive subspace tensor of order m. Suppose we run **LocalOpt** on $g(u)$, starting from a point u where $g(u) \geq \eta$, setting $\epsilon_1 \leq m\eta^{2/(m-2)}/M^{2/(m-2)}$. After $\mathrm{poly}(n, 1/\epsilon_1, \eta)$ iterations, we will have a point u^* where either $\|\pi_V(u^*)\| \geq 1 - 16\epsilon_1$ or $\|\pi_W(u^*)\| \geq 1 - 16\epsilon_1$.*

Proof. We proceed as in Lemma 4.2.1. u^* lies on a curve $C = \{s(u_V^*)^0 + t(u_W^*)^0 : s^2 + t^2 = 1, s \geq 0, t \geq 0\}$. We will show that neither s nor t is bounded away from 0 and 1.

Restricted to the curve $g(u) = g(s,t) = a_v s^m + a_w t^m$. Suppose that $a_w \leq 0$, then we must have that $s \geq (\eta/M)^{1/m}$. In this case, a direct calculation comparing $\langle Dg_u, u\rangle = ma_v s^{m-1} + ma_w t^{m-1}$ with $\|Dg_u\| = m\sqrt{a_v^2 s^{2(m-1)} + a_w^2 t^{2(m-1)}}$ will yield $s \geq 1 - 2\epsilon_1$. Thus, we may assume that both a_v and a_w are positive, and that $a_v \geq a_w$.

Suppose that for a unit vector u, we have $s, t \geq 16\sqrt{\epsilon_1}$, and the first-order gradient condition:

$$\frac{\langle Dg_u, u\rangle}{\|Dg_u\|} \geq 1 - \epsilon_1,$$

then,

$$\lambda_{\max}(D^2 g_u) \geq \frac{3m(m-1)g(u)}{4}$$

(where the eigenvalue is taken only in the subspace orthogonal to u). Thus, the algorithm continues making progress at such a vector u. To do this, we lower bound the top eigenvalue by the quadratic form in the direction $-tu_V^0 + su_W^0$, which is orthogonal to u.

$$\begin{aligned}
\lambda_{\max}(D^2 g_u) &\geq (-tu_V^0 + su_W^0)^T D^2 g_u (-tu_V^0 + su_W^0) \\
&= m(m-1)(a_v s^{m-2} t^2 + a_w s^2 t^{m-2}) \\
&= m(m-1)(a_v s^{m-1}, a_w t^{m-1}) \begin{pmatrix} t^2/s \\ s^2/t \end{pmatrix}
\end{aligned}$$

By construction, u has two nonzero coordinates, taking values s and t and all other coordinates zero. Dg_u has partial derivatives $ma_v s^{m-1}$ and $ma_w t^{m-1}$ in these directions. Thus,

$$\frac{\langle Dg_u, u \rangle}{\|Dg_u\|} \leq \frac{\begin{pmatrix} a_v s^{m-1} \\ a_w t^{m-1} \end{pmatrix}^T \begin{pmatrix} s \\ t \end{pmatrix}}{\left\| \begin{pmatrix} a_v s^{m-1} \\ a_w t^{m-1} \end{pmatrix} \right\|}$$

Thus we obtain the condition that:

$$\begin{pmatrix} a_v s^{m-1} \\ a_w t^{m-1} \end{pmatrix} = (1-\epsilon) \left\| \begin{pmatrix} a_v s^{m-1} \\ a_w t^{m-1} \end{pmatrix} \right\| \begin{pmatrix} s \\ t \end{pmatrix} + \sqrt{2\epsilon - \epsilon^2} \left\| \begin{pmatrix} a_v s^{m-1} \\ a_w t^{m-1} \end{pmatrix} \right\| r$$

where $0 \leq \epsilon \leq \epsilon_1$ and r is a unit vector orthogonal to (s,t). Substituting this into the

previous equation:

$$\begin{aligned}
\lambda_{\max}(D^2 g_u) &\geq m(m-1)\left[(1-\epsilon)\left\|\begin{pmatrix} a_v s^{m-1} \\ a_w t^{m-1}\end{pmatrix}\right\|\sqrt{2\epsilon-\epsilon^2}\left\|\begin{pmatrix} a_v s^{m-1} \\ a_w t^{m-1}\end{pmatrix}\right\| r^T \begin{pmatrix} t^2/s \\ s^2/t\end{pmatrix}\right] \\
&\geq m(m-1)\left\|\begin{pmatrix} a_v s^{m-1} \\ a_w t^{m-1}\end{pmatrix}\right\|\left((1-\epsilon)-\sqrt{2\epsilon-\epsilon^2}\left(\frac{1}{s}+\frac{1}{t}\right)\right) \\
&\geq m(m-1)\left\|\begin{pmatrix} a_v s^{m-1} \\ a_w t^{m-1}\end{pmatrix}\right\|\left((1-\epsilon)-2\sqrt{2\epsilon-\epsilon^2}\frac{1}{16\sqrt{\epsilon}}\right) \\
&\geq \frac{3m(m-1)g(u)}{4}
\end{aligned}$$

where the last estimate follows from the Cauchy-Schwartz inequality. □

4.2.5 Approximate tensors and approximate local optima

By using the robust Schwartz Zippel Lemma (Lemma 3.1.4) instead of the usual form, and **LocalOpt** at Lines 10 and 11 of **FindBasis**, we can obtain an efficient randomized algorithm. The major difficulty remaining is that we must bound the error incurred every time we call **LocalOpt**. The error analysis is technical: the idea is to obtain approximate versions of Lemmas 5.2.5 and 4.2.1, and to show that **LocalOpt** behaves well on these approximate versions. Consider the first iteration:

Lemma 4.2.5 (Two steps)**.** *Let T be an additive subspace tensor of order m. Let $u_1 = \sqrt{1-\delta}v_1 - \sqrt{\delta}w_1$ be the vector found in the first iteration of **FindBasis**, where v_1 and w_1 are unit vectors in V and W respectively. Suppose we run **LocalOpt** on $g(u)$ on the orthogonal subspace $u_1^{\perp}$, starting from a point u where $g(u) \geq \eta = M\delta^{1/16}$, setting $\epsilon_1 \leq \frac{m\eta^{2/(m-2)}}{M^{2/(m-2)}} - 60m^2M^2\delta^{5/16}$ as the error parameter in **LocalOpt**. After* $\mathrm{poly}(n, 1/\epsilon_1, \eta)$ *iterations, we will have a point u^* where either $\|\pi_V(u^*)\| \geq 1-\delta^{1/8}$ or $\|\pi_W(u^*)\| \geq 1-\delta^{1/8}$*

Tthe sequence of ideas in this proof is not unlike the proofs in Section 4.2: first we derive a nice representation of f (cf Lemma 5.2.5, then we analyse the support of a local optimum under this representation (cf Lemma 4.2.1) – we are not able to claim that the local optimum found is contained wholly in V or W, but since we are satisfied with approximate

local optima, we can bound the components around 0 and 1. All through this, we must bound the error, and try to push through the calculations of Lemma 4.2.4.

Proof of Lemma 4.2.5. First, we will construct an orthonormal basis which includes u_1: extend $\{v_1\}$ and $\{w_1\}$ to orthonormal bases $\{v_i\}$ and $\{w_i\}$ of V and W respectively. Replace v_1 and w_1 with the following two vectors:

$$u_1 = \sqrt{1-\delta}v_1 - \sqrt{\delta}w_1$$
$$\hat{u_1} = \sqrt{\delta}v_1 + \sqrt{1-\delta}w_1$$

Thus our basis will be $\{u_1, \hat{u}_1, v_2, \ldots, v_k, w_2, \ldots, w_l\}$. For a vector $x = (x_1, \ldots, x_n)$ in the basis of $\{v_i\}$ and $\{w_i\}$, we now have:

$$x = (\sqrt{1-\delta}x_1 - \sqrt{\delta}x_2, \sqrt{\delta}x_1 + \sqrt{1-\delta}x_2, x_3, \ldots, x_n)$$

which is simply a rotation (unitary transformation) in the first two coordinates.

We evaluate the m^{th} moment on the subspace orthogonal to u_1. Let ξ be a point on this orthogonal subspace: note that ξ has 0 component in the first coordinate:

$$\begin{aligned} f(\xi) &= T(\xi, \ldots, \xi) \\ &= T\left(\sqrt{\delta}\xi_2 e_1 + \sqrt{1-\delta}\xi_2 e_2 + \sum_{i=2}^{k} e_{v_i}\xi_{v_i} + \sum_{i=2}^{l} e_{w_i}\xi_{w_i}\right) \end{aligned}$$

Hence, we can apply Lemma 5.2.5: this gives a perturbed version of Lemma 5.2.5.

$$f(\xi) = \left(\delta\xi_2^2 + \sum_{i=2}^{k}\xi_{v_i}^2\right)^{m/2} T_V((\sqrt{\delta}\xi_2, \xi_{v_2}, \ldots, \xi_{v_k})^0) + \left((1-\delta)\xi_2^2 + \sum_{i=2}^{l}\xi_{w_i}^2\right)^{m/2} + T_W\left((\sqrt{1-\delta}\xi_2, \xi_{w_2}, \ldots, \xi_{w_l})^0\right)$$

Fixing a point $\xi^* \in u_1^{\perp} \cap \mathbb{S}^{n-1}$: we will give a curve C which passes through this point and remains on the unit sphere. We will analyse the value of the $g(\xi)$ on this curve – as before, every point which is a local optimum on $\mathbb{S}^{n-1}$ has to be a local optimum on C as

well. Thus by studying the local optima over C, we will be able to describe the strucure of the local optima in full space.

We may assume that all the ξ_i^* are nonnegative – otherwise we can pick simply negate the associated basis vector. We take the following as the components for C:

$$\xi_v^* = \frac{1}{\sqrt{\sum_{i=2}^k (\xi_{v_i}^*)^2}}(0, 0, \xi_{v_2}^*, \ldots, \xi_{v_k}^*, 0, \ldots, 0)$$
$$\xi_w^* = \frac{1}{\sqrt{(1-\delta)(\xi_2^*)^2 + \sum_{i=2}^l (\xi_{w_i}^*)^2}}(0, \sqrt{1-\delta}\xi_2^*, 0, \ldots, 0, \xi_{w_2}^*, \ldots, \xi_{v_l}^*)$$
$$\xi_1^* = (1, 0, \ldots, 0)$$

Since these are the only three directions that change along C, we will use these three vectors as an orthonormal basis. Now, defining the following quantity:

$$\alpha = (\xi_2^*)^2 / \left((1-\delta)(\xi_2^*)^2 + \sum_{i=2}^{l} (\xi_{w_i}^*)^2 \right)$$

we can write our curve C as:

$$C = \{y\xi_v^* + z\xi_w^* + \sqrt{\alpha\delta} z \xi_1^* : y^2 + (1+\delta\alpha)z^2 = 1, y, z \geq 0\}$$

Specifically, we will use the basis ξ_v^* and $(1+\alpha\delta)^{-1}(\xi_w^* + \xi_1^*)$. Note that in this basis, y is precisely the coordinate along the first basis vector and $(1+\delta\alpha)^{1/2}z$ is the coordinate along the second basis vector. Denote this latter quantity by z', then by the chain rule, we have that $\partial/\partial z' = (1+\delta\alpha)^{-1/2}\partial/\partial z$.

Restricted to C, the T_V and T_W terms simplify: note that $(\sqrt{1-\delta}\xi_2, \xi_{w_2}, \ldots, \xi_{w_l})^0$ remains constant on C, so the second term reduces to a constant, which we will denote with a_w. The first term does not remain constant, because there is an additional component in the direction of v_1, but this component always has a small magnitude. With a change of basis, we can simplify this expression to involving only y and z:

$$T_V((\sqrt{\delta}\xi_2, \xi_{v_2}, \ldots, \xi_{v_k})^0) = T_V' \left((\sqrt{\alpha\delta} z, y) \right)$$

(here the prime T_V' denotes the orthonormal basis change). We will denote the first term

by a_v. In full, our objective function on C is given by:

$$\begin{aligned} g(\xi) &= [\delta\alpha z^2 + y^2]^{(m/2)} T'_V \left((\sqrt{\alpha\delta} z, y) \right) + a_w z^m \\ &= [\delta\alpha z^2 + y^2]^{(m/2)} a_v(y, z) + a_w z^m \end{aligned}$$

Next we will examine the local optima on C: let ξ be the output of of **LocalOpt**: we will show that ξ has large projection with either the V or W subspace (cf Lemma 4.2.1). We will analyse the following cases:

1. $y^2 \leq \delta^{1/4}$ or $z^2 \leq \delta^{1/4}$.
2. $y^2 \geq \delta^{1/4}$ and $z^2 \geq 1/3$.
3. $z^2 \geq \delta^{1/4}$ and $y^2 \geq 1/3$.

Case 1: Suppose that $y^2 \leq \delta^{1/4}$, then we must have $z \geq \sqrt{1 - \delta^{1/4} - \alpha\delta}$. The approximate local optimum u that we compute has projection at least $\sqrt{1-\delta}$ on this local optimum, and hence, the projection of u onto w is at least:

$$\begin{aligned} \|\pi_W(u)\| &\geq \sqrt{(1-\delta)(1 - \delta^{1/4} - \alpha\delta)} - \sqrt{\delta} \\ &\geq 1 - \delta/2 - \delta^{1/4}/2 - \alpha\delta - \sqrt{\delta} \\ &\geq 1 - \delta^{1/4} \end{aligned}$$

In this case, for sufficiently small δ, we have:

$$\|\pi_W(u)\|^2 \geq 1 - \delta^{1/8}$$

The argument for when $z^2 \leq \delta^{1/4}$ is identical.

Case 2: We will prove that **LocalOpt** can not terminate in this region by carrying out the calculations of Lemma 4.2.4 whilst keeping track of errors. Thus, let ξ be a point in this range, we will show that if the first derivative condition in **LocalOpt** is satisfied, then the second derivative condition is unsatisfied, thus **LocalOpt** can not terminate at ξ. First, let us examine how f changes over C:

Claim 4.2.6 (First partials under perturbations)**.** *In the range where* $y^2 \geq \delta^{1/4}$ *and* $z^2 \geq 1/3$*,*

$$\left|\frac{\partial g}{\partial y} - ma_v y^{m-1}\right| \leq 3mM\sqrt{\delta}$$
$$\left|\frac{\partial g}{\partial z} - ma_w z^{m-1}\right| \leq 4mM\sqrt{\delta}$$

As a corollary, via the triangle inequality, we have that:

$$\|(g_y, g_z)\| \geq m \left\|(a_v y^{m-1}, a_w z^{m-1})\right\| - 5mM\sqrt{\delta}$$

Claim 4.2.7 (Second partials under perturbations)**.** *In the range where* $y^2 \geq \delta^{1/4}$ *and* $z^2 \geq 1/3$*:*

$$\left|\frac{\partial^2 g}{\partial y^2} - m(m-1)a_v y^{m-2}\right| \leq c_{vv} m^2 M\sqrt{\delta}$$
$$\left|\frac{\partial^2 g}{\partial z^2} - m(m-1)a_w z^{m-2}\right| \leq c_{ww} m^2 M\sqrt{\delta}$$
$$\left|\frac{\partial^2 g}{\partial y \partial z}\right| \leq c_{vw} m^2 M\sqrt{\delta}$$

where c_{vv}*,* c_{vw} *and* c_{ww} *are absolute constants bounded by 20.*

Throughout the rest of this calculation, we will use the basis of $n-1$ vectors consisting of $\{\xi_v^*,\ (1+\alpha\delta)^{-1}(\xi_w^* + \xi_1^*)\}$, and any orthonormal extension to $u_1^\perp$. In particular, in this basis, $\xi = (y, z', 0, \ldots, 0)$.

As before, we will lower bound the contribution of the second derivative term. Our direction of movement will be $(-z', y, 0, \ldots, 0)$. This vector is clearly a unit vector orthogonal to ξ. where top eigenvalue is taken orthogonal to ξ.

$$\begin{aligned}\lambda_{\max}(D^2 g_\xi) &\geq (-z', y) D^2 g_\xi \begin{pmatrix} -z' \\ y \end{pmatrix} \\ &\geq (-\sqrt{1+\alpha\delta}z, y) \begin{pmatrix} g_{yy} & g_{z'y} \\ g_{z'y} & g_{z'z'} \end{pmatrix} \begin{pmatrix} -\sqrt{1+\delta\alpha}z \\ y \end{pmatrix}\end{aligned}$$

We can further use Claim 4.2.7 to simplify the other components of the quadratic form:

$$\begin{aligned}\lambda_{\max}(D^2 g_\xi) &\geq (1+\delta\alpha) z^2 g_{yy} + y^2 g_{z'z'} - 2c_{vw} m^2 M\sqrt{\delta} \\ &\geq (1+\delta\alpha) m(m-1)(a_v y^{m-1}, a_w z^{m-1}) \begin{pmatrix} z^2/y \\ y^2/z \end{pmatrix} - (1+\delta\alpha)(c_{zz} + c_{yy} + 2c_{zw}) m^2 M\sqrt{\delta}\end{aligned}$$

Our first derivative condition is given by:

$$\frac{\langle Dg_\xi, \xi\rangle}{\|Dg_\xi\|} \geq 1 - \epsilon_1$$

Since $\xi = (y, z', 0, \ldots, 0)$ has only two nonzero components, we need only evaluate two components of the derivative: furthermore, we can lower bound the norm $\|Dg_\xi\| \geq \|(g_y, g_{z'})\|$, which gives the following lower bound:

$$\frac{(g_y, g_{z'}) \begin{pmatrix} y \\ z' \end{pmatrix}}{\|(g_y, g_{z'})\|} \geq 1 - \epsilon_1$$

Rearranging, and applying Claim 4.2.6 yields:

$$\begin{aligned} m(a_v y^{m-1}, a_w z^{m-1}) \begin{pmatrix} y \\ z \end{pmatrix} &\geq (1-\epsilon_1) \|(g_y, g_{z'})\| - 7mM\sqrt{\delta} \\ &\geq m(1-\epsilon_1) \left\|(a_v y^{m-1}, a_w z^{m-1})\right\| - 12mM\sqrt{\delta} \\ &\geq m(1 - \epsilon_1 - \frac{12M\sqrt{\delta}}{\eta}) \left\|(a_v y^{m-1}, a_w z^{m-1})\right\| \end{aligned}$$

Thus, we can rewrite this relationship for unit vector r orthogonal to $(a_v y^{m-1}, a_w z^{m-1})$ and $0 \leq \epsilon \leq \epsilon_1 + \frac{12M\sqrt{\delta}}{\eta}$:

$$\begin{pmatrix} a_v y^{m-1} \\ a_w z^{m-1} \end{pmatrix} = (1-\epsilon) \left\|(a_v y^{m-1}, a_w z^{m-1})\right\| \begin{pmatrix} y \\ z \end{pmatrix} + \sqrt{2\epsilon - \epsilon^2} \left\|(a_v y^{m-1}, a_w z^{m-1})\right\| r$$

Substituting this back into our lower bound for $\lambda_{\max}$ yields:

$$\begin{aligned}\lambda_{\max} &\geq (1+\delta\alpha)(1-\epsilon) \left\|(a_v y^{m-1}, a_w z^{m-1})\right\| (z^2 + y^2) - \sqrt{2\epsilon - \epsilon^2} \left\|(a_v y^{m-1}, a_w z^{m-1})\right\| \left(\frac{1}{y} + \frac{1}{z}\right) \\ &\quad - 80m^2 M\sqrt{\delta} \\ &\geq (1+\delta\alpha)(1-\delta^{1/6}) m(m-1) f(\xi) - \sqrt{2}\delta^{1/24} - 80m^2 M\sqrt{\delta} \\ &\geq \frac{3}{4} m(m-1) f(\xi)\end{aligned}$$

where we used the Cauchy-Schwartz inequality for:

$$\left\|(a_v y^{m-1}, a_w z_m - 1)\right\| \geq a_v y^m + a_w z^m \geq g(\xi) - mM\sqrt{\alpha\delta}$$

Case 3: This case follows from the exactly the same analysis as above. It is in fact substantially easier, as the denominator terms $\alpha\delta z + y$ are in fact all bounded by constants now, and hence the numerator is small enough in almost all cases above to bound the terms.

We now provide the proofs for the claims regarding the coefficients a_v and a_w. In explicitly taking derivatives, it is important to note the following:

$$\begin{aligned} a_v &= T_V'\left((\sqrt{\alpha\delta}z, y)\right) \\ &= \frac{1}{(\alpha\delta z^2 + y^2)^{(m/2)}} T_V'\left((\sqrt{\alpha\delta}z, y)\right) \end{aligned}$$

For ease of notation, denote $\phi = (\sqrt{\alpha\delta}z, y)$, we will suppress all but one ϕ argument in our moment tensors, thus we will write $A(\phi)$ instead of $A(\phi, \ldots, \phi)$, and $A(\phi, e_1)$ instead of $A(\phi, \ldots, \phi, e_1)$. If A is a m^{th} order tensor, its derivative has components given by $(D\hat{A}_\phi)_i = mA(\phi, e_i)$ where A takes $(m-1)$ copies of ϕ. We also have the Hessian D^2: $(D^2 A_\phi)_{ij} = m(m-1)A(\phi, e_i, e_j)$. We can bound the spectral norm of $D^2 A$ using Claim 3.2.10, which yields $\lambda_{max}(D^2 A) \leq m(m-1)M$.

Proof of Claim 4.2.6. Firstly, we have:

$$\begin{aligned} \frac{\partial g}{\partial y} &= my(\delta\alpha z^2 + y^2)^{(m/2)-1} a_v + (\delta\alpha z^2 + y^2)^{(m/2)} \frac{\partial a_v}{\partial y} \\ \frac{\partial g}{\partial z} &= mz^{m-1} a_w + m\alpha\delta z(\delta\alpha z^2 + y^2)^{(m/2)-1} a_v + (\delta\alpha z^2 + y^2)^{(m/2)} \frac{\partial a_v}{\partial z} \end{aligned}$$

The $m\alpha\delta z(\delta\alpha z^2 + y^2)^{(m/2)-1} a_v$ is upper bounded in absolute value in $mM\delta$. Similarly, it is also clear that:

$$\left| my(\delta\alpha z^2 + y^2)^{(m/2)-1} a_v - m a_v y^{m-1} \right| \leq mM\sqrt{\delta}$$

Thus it remains to show that the partial derivative terms are small:

$$\begin{aligned} (\delta\alpha z^2 + y^2)^{(m/2)} \frac{\partial a_v}{\partial y} &= (\delta\alpha z^2 + y^2)^{(m/2)} \left[\frac{-my}{(\delta\alpha z^2 + y^2)^{(m/2)+1}} A(\phi, \phi) + \frac{m}{(\delta\alpha z^2 + y^2)^{(m/2)}} A(\phi, e_1) \right] \\ &= m\left(\frac{-y\sqrt{\alpha\delta}zA(\phi, e_2) - y^2 A(\phi, e_1)}{\alpha\delta z^2 + y^2} + A(\phi, e_1) \right) \\ &= m\frac{-y\sqrt{\alpha\delta}zA(\phi, e_2) + \alpha\delta z^2 A(\phi, e_1)}{\alpha\delta z^2 + y^2} \end{aligned}$$

When we have a term like $A(\phi, \ldots, \phi, e_1)$, the arguments are not normalised. In particular:

$$A(\phi, \ldots, \phi, e_1) = (\delta\alpha z^2 + y^2)^{(m-1)/2} A(\phi^0, \ldots, \phi^0, e_1)$$

Thus, normalising gives:

$$\begin{aligned}\left|(\delta\alpha z^2 + y^2)^{(m/2)} \frac{\partial a_v}{\partial y}\right| &\leq mM\sqrt{\delta} + m\delta M \\ &\leq 2mM\sqrt{\delta}\end{aligned}$$

For the other partial derivative, we want to compute:

$$\begin{aligned}(\delta\alpha z^2 + y^2)^{(m/2)} \frac{\partial a_v}{\partial z} &= (\delta\alpha z^2 + y^2)^{(m/2)} \left[\frac{-m\alpha\delta z}{(\alpha\delta z^2 + y^2)^{(m/2)+1}} A(\phi, \phi) + \frac{m\sqrt{\alpha\delta}}{(\alpha\delta z^2 + y^2)^{m/2}} A(\phi, e_2)\right] \\ &= m\sqrt{\alpha\delta}\left(\frac{-\sqrt{\alpha\delta} z A(\phi, \phi) + \alpha\delta z^2 A(\phi, e_2) + y^2 A(\phi, e_2)}{\alpha\delta z^2 + y^2}\right)\end{aligned}$$

Applying the same method:

$$\left|(\delta\alpha z^2 + y^2)^{(m/2)} \frac{\partial a_v}{\partial z}\right| \leq 3mM\sqrt{\delta}$$

Hence combining this with our earlier bound, we have the desired inequality. □

Proof of Claim 4.2.7. By direct calculation, we obtain:

$$\frac{\partial^2 g}{\partial y^2} = (\delta\alpha z^2 + y^2)^{(m/2)} \frac{\partial^2 a_v}{\partial y^2} + 2my(\delta\alpha z^2 + y^2)^{(m/2)-1} \frac{\partial a_v}{\partial y} + m a_v (\delta\alpha z^2 + y^2)^{(m/2)-2} (\delta\alpha z^2 + (m-1)y^2)$$

We now estimate the three terms in this sum – the first two terms will be of order $\sqrt{\delta}$, and the last term will give us approximately $m(m-1)a_v y^{m-2}$.

$$\begin{aligned}&(\delta\alpha z^2 + y^2)^{(m/2)} \frac{\partial^2 a_v}{\partial y^2} \\ &= (\delta\alpha z^2 + y^2)^{(m/2)} \left\{\frac{-m^2 y}{(\alpha\delta z^2 + y^2)^{(m/2)+1}} \left[\frac{-y\sqrt{\alpha\delta} z A(\phi, e_2)}{\alpha\delta z^2 + y^2} + \frac{\alpha\delta z^2 A(\phi, e_1)}{\alpha\delta z^2 + y^2}\right] + \frac{m}{(\alpha\delta z^2 + y^2)^{m/2}} \left[\frac{-\sqrt{\alpha\delta} z A}{\alpha\delta z^2 -}\right.\right. \\ &\left.\left. + \frac{-(m-1)y\sqrt{\alpha\delta} z A(e_2, e_1)}{\alpha\delta z^2 + y^2} + \frac{y^2\sqrt{\alpha\delta} z A(\phi, e_2)}{(\alpha\delta z^2 + y^2)^2} + \frac{\alpha\delta z^2 (m-1) A(e_1, e_1)}{\alpha\delta z^2 + y^2} + \frac{-2y\alpha\delta z^2 A(\phi, e_1)}{(\alpha\delta z^2 + y^2)^2}\right]\right\} \\ &= (-m^2 y)\left[\frac{-y\sqrt{\alpha\delta} z A(\phi, e_2)}{(\alpha\delta z^2 + y^2)^2} + \frac{\alpha\delta z^2 A(\phi, e_1)}{(\alpha\delta z^2 + y^2)^2}\right] + m\left[\frac{-\sqrt{\alpha\delta} z A(\phi, e_2)}{\alpha\delta z^2 + y^2} + \frac{-(m-1)y\sqrt{\alpha\delta} z A(e_2, e_1)}{\alpha\delta z^2 + y^2}\right. \\ &\left. + \frac{y^2\sqrt{\alpha\delta} z A(\phi, e_2)}{(\alpha\delta z^2 + y^2)^2} + \frac{\alpha\delta z^2 (m-1) A(e_1, e_1)}{\alpha\delta z^2 + y^2} + \frac{-2y\alpha\delta z^2 A(\phi, e_1)}{(\alpha\delta z^2 + y^2)^2}\right]\end{aligned}$$

We will bound the magnitude of every term in this sum. Consider the first term of the form:

$$\left|(-m^2 y)\frac{-y\sqrt{\alpha\delta}zA(\phi,e_2)}{(\alpha\delta z^2+y^2)^2}\right| \leq m^2\left|\frac{y^2\sqrt{\delta}A(\phi,e_2)}{(\alpha\delta z^2+y^2)^2}\right|$$

Thus, since $m \geq 3$:

$$\left|(-m^2 y)\frac{-y\sqrt{\alpha\delta}zA(\phi,e_2)}{(\alpha\delta z^2+y^2)^2}\right| \leq 3m^2 M\left|\frac{y^2\sqrt{\delta}}{\alpha\delta z^2+y^2}\right|$$

Now, $y^2/(\alpha\delta z^2+y^2) \leq 1$, hence:

$$\left|(-m^2 y)\frac{-y\sqrt{\alpha\delta}zA(\phi,e_2)}{(\alpha\delta z^2+y^2)^2}\right| \leq 3m^2 M\sqrt{\delta}$$

Of the seven terms in the sum, the first, third and fifth terms can be analysed exactly as above, and their sum can be upper bounded by $15m^2M\sqrt{\delta}$. For the remaining terms we have to use our lower bound on y, for example:

$$\begin{aligned}\left|(-my)\frac{\alpha\delta z^2 A(\phi,e_1)}{(\alpha\delta z^2+y^2)^2}\right| &\leq mM\left|\frac{\delta y}{\alpha\delta z^2+y^2}\right| \\ &\leq mM\left|\frac{\delta}{y}\right| \\ &\leq mM\delta^{7/8}\end{aligned}$$

By this reasoning, we can bound all seven terms by $m^2M\sqrt{\delta}$, hence this term in $\partial^2 g/\partial y^2$ contributes is bounded in absolute value by $7m^2M\sqrt{\delta}$. For the second term in that expression, the analysis is almost identical to the previous claim and gives

$$\begin{aligned}\left|2my(\delta\alpha z^2+y^2)^{(m/2)-1}\frac{\partial a_v}{\partial y}\right| &= 2m^2\left|y(\delta\alpha z^2+y^2)^{(m/2)-1}\frac{(-y\sqrt{\alpha\delta}zA(\phi^0,e_2)+\alpha\delta z^2A(\phi^0,e_1))}{(\delta\alpha z^2+y^2)^{3/2}}\right| \\ &\leq 2m^2\left|y\frac{(-y\sqrt{\alpha\delta}zA(\phi^0,e_2)+\alpha\delta z^2A(\phi^0,e_1))}{(\delta\alpha z^2+y^2)}\right| \\ &\leq 2m^2M\sqrt{\delta}+2m^2M\left|\frac{\delta}{y}\right| \\ &\leq 4m^2M\sqrt{\delta}\end{aligned}$$

Thus, we have:

$$\left|\frac{\partial^2 g}{\partial y^2}-ma_v(\delta\alpha z^2+y^2)^{(m/2)-2}(\delta\alpha z^2+(m-1)y^2)\right| \leq 19m^2M\sqrt{\delta}$$

By applying the triangle inequality:

$$\left|ma_v(\delta\alpha z^2 + y^2)^{(m/2)-2}(\delta\alpha z^2 + (m-1)y^2) - m(m-1)a_v y^{m-2}\right| \leq m^2 M\sqrt{\delta}$$

Thus we have the desired result for the second partial with respect to y. The other second derivatives are computed in a similar way. □

□

Using the above, we are now examine what happens after t iterations of **FindBasis**. The following theorem shows that after k iterations of **FindBasis**, our error blows up at most doubly exponentially in k. The proof holds for **ExtendBasis** is as well.

Theorem 4.2.8 (Multiple iterations)**.** *Suppose **FindBasis** finds $j \leq k$ orthogonal vectors $\{u_1, \ldots, u_j\}$ of $g(u)$ taking ϵ_1 such that $\eta \leq M\epsilon_1^{1/16^j}$ for each call of **LocalOpt**, then $\|\pi_V(u_j)\|^2 \geq 1 - \epsilon_1^{(1/16)^j}$.*

Proof of Theorem 4.2.8. After t iterations, we have a basis of orthonormal vectors $\{u_1, \ldots, u_t\}$ where each u_i is close to some vector in V:

$$\begin{aligned}
u_1 &= a_{11}v_1 + b_{11}w_1 \\
u_2 &= a_{21}v_1 + a_{22}v_2 + b_{21}w_1 + b_{22}w_2 \\
&\vdots \quad \vdots \\
u_t &= a_{t1}v_1 + \cdots + a_{tt}v_t + b_{t1}w_1 + \cdots b_{tt}w_t
\end{aligned}$$

We use the orthonormal basis $\{u_i\}$, $\{v_{t+1}, \ldots, v_k\}$, the remaining vectors in W $\{w_{t+1}, \ldots, w_{n-k}\}$, and approximate copies of $\{w_1, \ldots, w_t\}$. This last set is given by:

$$\begin{aligned}
w_1' &= c_1 w_1 + \sum_{i=1}^{t} d_{1i}v_i + \sum_{i=1}^{t} e_{1i}w_i \\
&\vdots \quad \vdots \\
w_t' &= c_t w_t + \sum_{i=1}^{t} d_{ti}v_i + \sum_{i=1}^{t} e_{ti}w_i
\end{aligned}$$

In these sums we have $d_{ii} = e_{ii} = 0$, and we have orthonormality between these vectors. Consider the explicit coordinates of ξ:

$$\xi = (\sum_{i=1}^{t} \xi_{w_i'} d_{i1}, \ldots, \sum_{i=1}^{t} \xi_{w_i'} d_{it}, \xi_{v_{t+1}}, \ldots, \xi_{v_k}, \xi_{w_1'} c_1 + \sum_{i=1}^{t} \xi_{w_i'}' e_{i1}, \xi_{w_t'} c_t + \sum_{i=1}^{t} \xi_{w_i'} e_{it}, \xi_{w_{t+1}}, \ldots, \xi_{w_{n-k}})$$

The two vectors ξ_V and ξ_W formed from ξ have total norm 1. Now, we can apply Lemma 5.2.5, to obtain:

$$f_m(\xi') = \left(\sum_{j=t+1}^{k} \xi_{v_j}^2 + \sum_{j=1}^{t} \left(\sum_{i=1}^{t} \xi_{w_i'} d_{ij} \right)^2 \right)^{m/2} a_v + \left(\sum_{j=t+1}^{n-k} \xi_{w_j}^2 + \sum_{j=1}^{t} \left(\xi_{w_j'} c_j + \sum_{i=1}^{t} \xi_{w_i'} e_{ij} \right)^2 \right)^{m/2} a_w$$

where the expectation term a_v is given by:

$$a_v = T_V \left((\sum_{i=1}^{t} \xi_{w_i'} d_{i1}, \ldots, \sum_{i=1}^{t} \xi_{w_i'} d_{it}, \xi_{v_{t+1}}, \ldots, \xi_{v_k})^0 \right)$$

(and similarly for a_w). As in the single iteration case, we restrict to a curve. Fix an output ξ^* of **FindBasis**: we will fix the ratio of the components $\{\xi_{w_j}\}$ in the ratio of ξ^*, and similarly, we will fix the ratios of $\{\xi_{w_1'}, \ldots, \xi_{w_t'}, \xi_{w_{t+1}}, \ldots, \xi_{w_{n-k}}\}$ according to ξ^* as well. This gives the following restriction on our curve:

$$g(\xi') = a_v \left[(y')^2 + (z')^2 \left(\sum_{j=1}^{t} \left(\sum_{i=1}^{t} d_{ij} \xi_{w_i'}^* / l \right)^2 \right) \right]^{m/2} + a_w (z')^m$$

where l is a constant given by:

$$l = \frac{1}{\left(\sum_{j=t+1}^{n-k} (\xi_{w_j}^*)^2 + \sum_{j=1}^{t} \left(\xi_{w_j'}^* c_j + \sum_{i=1}^{t} \xi_{w_i'}^* e_{ij} \right)^2 \right)}$$

The coefficient of z'^2 is bounded by at most $2t(\epsilon_1^{1/16})^t$, hence using the previous lemma for a single iteration, the output produced here is a $(t+1)^{th}$ vector u_{t+1} such that:

$$\begin{aligned} \langle u_{t+1}, u^* \rangle &\geq 1 - \left(2t(\epsilon_1^{1/16})^t \right)^{1/8} \\ &\geq 1 - (\epsilon_1^{1/16})^{t+1} \end{aligned}$$

for sufficiently small ϵ_1 (relative to k). □

4.3 Decomposition into rank 1 components

The following is based on [65].

Our goal, in analogy with spectral decomposition for matrices, is to recover (symmetric) rank-1 decompositions of tensors. Unfortunately, there are no known algorithms with provable guarantees when $m > n$, and in fact this problem is NP-hard in general [33, 72]. It is an open research question to characterize, or even give interesting sufficient conditions, for when a rank-1 decomposition of a tensor T as in (4) is unique and computationally tractable. For the case $d = 2$, a necessary and sufficient condition for uniqueness is that the eigenvalues of T are distinct. Indeed, when eigenvalues repeat, rotations of the A_i in the degenerate eigensubspaces with repeated eigenvalues lead to the same matrix M.

For $d > 2$, if the A_i are orthogonal, then the expansion in (4) is unique and can be computed efficiently. The algorithm is power iteration that recovers one A_i at a time (see e.g. [9]). The requirement that the A_i are orthogonal is necessary for this algorithm, but if one also has access to the order-2 tensor (i.e., matrix) in addition, $M = \sum_{i=1}^{m} A_i \otimes A_i$, and the A_i are linearly indepenent, then one can arrange for the orthogonality of the A_i by a suitable linear transformation. However, the fundamental limitation remains that we must take $m \leq n$ simply because we can not have more than n orthogonal vectors in $\mathbb{R}^n$. The main result of this section is that we are able to give a robust polynomial time algorithm for decomposing certain order r tensors into m rank 1 components, where m can be as high as $O(n^{r/2})$, which is a dramatic improvement on $O(n)$ components in the earlier work. Subsequent to our work, other authors have studied the technical requirement we need for our algorithm to work, and have shown that in a smoothed analysis setting that the technical condition that we require for our decomposition holds [25].

Here, by considering a slightly modified setting where we are allowed some additional information, we are able to lever this into an algorithm for the general case: suppose we have access to two tensors, both of order d, which share the same rank-1 components, but have different coefficients:

$$T_\mu = \sum_{i=1}^{m} \mu_i A_i^{\otimes d}, \qquad T_\lambda = \sum_{i=1}^{m} \lambda_i A_i^{\otimes d}.$$

Given such a *pair* of tensors T_μ and T_λ, can we recover the rank-1 components A_i?

We answer this question in the affirmative for even orders $d \in 2\mathbb{N}$, and give a provably good algorithm for this problem assuming that the ratios μ_i/λ_i are distinct. Additionally, we assume that the A_i are not scalar multiples of each other, a necessary assumption. We make this quantitative via the m^{th} singular value of the matrix with columns given by $A_i^{\odot d/2}$.

Our main idea here is that we do not attempt to decompose a single tensor into its rank-1 components. This is an NP-hard problem in general, and to make it tractable, previous work uses additional informaton and structural assumptions or places strong restrictions on how large m can be as a function of n. Instead, we consider *two* tensors which share the same rank-1 components and compose the tensors in a specific way, thereby extracting the desired rank-1 components. In the following $\text{vec}\left(A_i^{\otimes d/2}\right)$ denotes the tensor $A_i^{\otimes d/2}$ flattened into a vector. The algorithm's input consists of: tensors T_μ, T_λ, and parameters $n, m, d, \Delta, \epsilon$ as explained in the following theorem.

Theorem 4.3.1 (Tensor decomposition)**.** *Let A be an $n \times m$ matrix with $m > n$ and columns with unit norm, and let $T_\mu, T_\lambda \in \mathbb{R}^{n \times \cdots \times n}$ be order d tensors such that $d \in 2\mathbb{N}$ and*

$$T_\mu = \sum_{i=1}^{m} \mu_i A_i^{\otimes d} \qquad T_\lambda = \sum_{i=1}^{m} \lambda_i A_i^{\otimes d},$$

where $\text{vec}\left(A_i^{\otimes d/2}\right)$ are linearly independent, $\mu_i, \lambda_i \neq 0$ and $\left|\frac{\mu_i}{\lambda_i} - \frac{\mu_j}{\lambda_j}\right| > \Delta$ for all i, j and $\Delta > 0$. Then, algorithm TensorDecomposition(T_μ, T_λ) outputs vectors $A'_1, \ldots, A'_m$ with the following property. There is a permutation $\pi : [m] \to [m]$ and signs $\alpha : [m] \to \{-1, 1\}$ such that for $i \in [m]$ we have

$$\left\| \alpha_i A'_{\pi(i)} - A_i \right\|_2 \leq \epsilon.$$

The running time is $\text{poly}\left(n^d, \frac{1}{\epsilon}, \frac{1}{\Delta}, \frac{1}{\sigma_{\min}(A^{\odot d/2})}\right)$.

The polynomial in the running time above can be made explicit. It basically comes from the time complexity of SVD and eigenvector decomposition of diagonalizable matrices. We note that in contrast to previous work on tensor decompositions [68, 53, 39, 119], our

method has provable finite sample guarantees. We give a robust version of the above, stated as Theorem 4.3.5.

As a core subroutine for all problems above, we develop a general theory of efficient tensor decompositions for pairs of tensors, which allows us to recover a rank-1 tensor decomposition from two homogeneous tensor relations. As noted in the literature, such a pair of tensor equations can be obtained from one tensor equation by applying two random vectors to the original equation, losing one in the order of the tensor. Our tensor decomposition "flattens" these tensors to matrices and performs an eigenvalue decomposition. The matrices in question are not Hermitian or even normal, and hence we use more general methods for eigendecomposition (in particular, tensor power iterations cannot be used to find the desired decompositions). The algorithm for tensor decomposition via simultaneous tensor diagonalization is essentially due to Leurgans et al [95]; to the best of our knowledge, ours is the first robust analysis.

In subsequent work, Bhaskara et al. [25] have sketched a similar robustness analysis with a different application.

4.3.1 Algorithm

Our algorithm works by flattening tensors T_μ and T_λ into matrices M_μ and M_λ which have the following form:

$$M_\mu = (A^{\odot d/2})\mathrm{diag}\,(\mu_i)\,(A^{\odot d/2})^T, \qquad M_\lambda = (A^{\odot d/2})\mathrm{diag}\,(\lambda_i)\,(A^{\odot d/2})^T.$$

Taking the product $M_\mu M_\lambda^{-1}$ yields a matrix whose eigenvectors are the columns of $A^{\odot d/2}$ and whose eigenvalues are μ_i/λ_i:

$$\begin{aligned} M_\mu M_\lambda^{-1} &= (A^{\odot d/2})\mathrm{diag}\,(\mu_i)\,(A^{\odot d/2})^T((A^{\odot d/2})^T)^{-1}\mathrm{diag}\,(\lambda_i)^{-1}\,(A^{\odot d/2})^{-1} \\ &= (A^{\odot d/2})\mathrm{diag}\,(\mu_i/\lambda_i)\,(A^{\odot d/2})^{-1}. \end{aligned}$$

This is the essential intuition for our algorithm. The rest of this section simply hardens the algorithm against input noise by carefully applying spectral perturbation bounds (with an added complication that spectral decompositions may not exist for some of our matrices).

Actually, for the last equation to make sense one needs that $A^{\odot d/2}$ be invertible which will generally not be the case as $A^{\odot d/2}$ is not even a square matrix in general. We handle this by restricting M_μ and M_λ to linear transform from their pre-image to the image. This is the reason for introducing matrix W in algorithm **Diagonalize**(M_μ, M_λ) below.

The main algorithm below is **Tensor Decomposition**(T_μ, T_λ) which flattens the tensors and calls subroutine **Diagonalize**(M_μ, M_λ) to get estimates of the $A_i^{\odot d/2}$, and from this information recovers the A_i themselves. In our application it will be the case that $\mu, \lambda \in \mathbb{C}^m$ and $A_i \in \mathbb{R}^n$. The discussion below is tailored to this situation; the other interesting cases where everything is real or everything is complex can also be dealt with with minor modifications.

Diagonalize(M_μ, M_λ)

1. Compute the SVD of $M_\mu = V\Sigma U^T$, and let W be the left singular vectors (columns of V) corresponding to the m largest singular values. Compute the matrix $M = (W^T M_\mu W)(W^T M_\lambda W)^{-1}$.
2. Compute the eigenvector decomposition $M = PDP^{-1}$.
3. Output columns of WP.

Tensor Decomposition(T_μ, T_λ)

1. Flatten the tensors to square matrices to obtain $M_\mu = \tau^{-1}(T_\mu)$ and $M_\lambda = \tau^{-1}(T_\lambda)$.
2. $WP = Diagonalize(M_\mu, M_\lambda)$.
3. For each column C_i of WP, let $C_i' := \mathrm{Re}\left(e^{i\theta^*} C_i\right) / \left\|\mathrm{Re}\left(e^{i\theta^*} C_i\right)\right\|$ where $\theta^* = \mathrm{argmax}_{\theta \in [0,2\pi]} \left(\left\|\mathrm{Re}\left(e^{i\theta} C_i\right)\right\|\right)$.
4. For each column C_i', let $v_i \in \mathbb{R}^n$ be such that $v_i^{\otimes d/2}$ is the best rank-1 approximation to $\tau(C_i')$.

The columns $C_i = WP_i$ are eigenvectors computed in subroutine **Diagonalize**. Ideally, we would like these to equal $A_i^{\odot d/2}$. We are going to have errors introduced because of sampling, but in addition, since we are working in the complex field we do not have control

over the phase of C_i (the output of **Diagonalize** obtained in Step 3 of **Tensor Decomposition**), and for $\rho \in \mathbb{C}$ with $|\rho| = 1$, ρC_i is also a valid output of **Diagonalize**. In Step 3 of **Tensor Decomposition**, we recover the correct phase of $C_i \in \mathbb{C}^n$ which will give us a vector in $C_i' \in \mathbb{R}^n$. We do this by choosing the phase maximizing the norm of the real part.

In Step 4, we have $v^{\otimes d} + E$, where E is an error tensor, and we want to recover v. We can do this approximately when $\|E\|_F$ is sufficiently small just by reading off a one-dimensional slice (e.g. a column in the case of matrices) of $v^{\otimes d} + E$ (say the one with the maximum norm).

For the computation of eigenvectors of diagonalizable (but not normal) matrices over the complex numbers, we can employ any of the several algorithms in the literature (see for example [64, 112] for a number of algorithms used in practice). In general, these algorithms employ the same atomic elements as the normal case (Jacobi iterations, Householder transformations etc.), but in more sophisticated ways. The perturbation analysis of these algorithms is substantially more involved than for normal matrices; in particular, it is not necessarily the case that a (small) perturbation to a diagonalizable matrix results in another diagonalizable matrix. We contend with all these issues in Section 4.3.3. In particular we note that while *exact* analysis is relatively straightforward (Theorem 4.3.4), a robust version that recovers the common decomposition of the given tensors takes considerable additional care (Theorem 4.3.5).

In Step 3 of Tensor Decomposition, we get an approximation of $v_i^{\odot d/2}$ up to a phase factor. We first correct the phase by maximizing the projection onto $\mathbb{R}^n$. To this end we prove

Lemma 4.3.2. *Let $v \in \mathbb{C}^n$ and $u \in \mathbb{R}^n$ be unit vectors such that for some $\varphi \in [0, 2\pi]$ we have $\left\|e^{i\varphi}v - u\right\| \leq \epsilon$ for $0 \leq \epsilon \leq 1/2$. Let $\theta^* = \operatorname{argmax}_{\theta\in[0,2\pi]}(\left\|\operatorname{Re}\left(e^{i\theta}v\right)\right\|)$ and $u' = \operatorname{Re}\left(e^{i\theta^*}v\right) / \left\|\operatorname{Re}\left(e^{i\theta^*}v\right)\right\|$. Then there is a sign $\alpha \in -1, 1$ such that*

$$\left\|\alpha u - u'\right\| \leq 11\sqrt{\epsilon}.$$

Proof. Without loss of generality, we will assume that $\varphi = 0$, hence $\|v - u\| \leq \epsilon$. By the

optimality of θ^*

$$\left\|\mathrm{Re}\left(e^{i\theta^*}v\right)\right\| \geq \|\mathrm{Re}\,(v)\| \geq 1-\epsilon.$$

Let us denote $v' = e^{i\theta^*}v$, then we have $\|\mathrm{Re}\,(v')\|^2 + \|\mathrm{Im}\,(v')\|^2 = 1$ which implies that $\|\mathrm{Im}\,(v')\|^2 \leq 2\epsilon - \epsilon^2 < 2\epsilon$. Now using $\epsilon \leq 1/2$ we have

$$\begin{aligned}
\|v' - u'\| &\leq \|\mathrm{Re}\,(v') - u'\| + \|\mathrm{Im}\,(v')\| \\
&= \left\|\mathrm{Re}\,(v') - \frac{\mathrm{Re}\,(v')}{\|\mathrm{Re}\,(v')\|}\right\| + \|\mathrm{Im}\,(v')\| \\
&\leq \|\mathrm{Re}\,(v')\|\left(\frac{1}{1-\epsilon} - 1\right) + \|\mathrm{Im}\,(v')\| \\
&\leq 2\epsilon + \sqrt{2\epsilon} \leq 4\sqrt{\epsilon},
\end{aligned}$$

and

$$\left\|u' - e^{i\theta^*}u\right\| \leq \left\|u' - e^{i\theta^*}v\right\| + \left\|e^{i\theta^*}v - e^{i\theta^*}u\right\| = \|u' - v'\| + \|u - v\| < 5\sqrt{\epsilon}.$$

This implies $\left|\mathrm{Re}\left(e^{i\theta^*}\right)\right| \geq 1 - 5\sqrt{\epsilon}$. Hence there is a sign $\alpha \in -1, 1$ such that $\left|e^{i\theta^*} - \alpha\right| \leq 10\sqrt{\epsilon}$ (we omit some routine computations). Finally,

$$\|u' - \alpha u\| \leq \left\|u' - e^{i\theta^*}u\right\| + \left\|e^{i\theta^*}u - \alpha u\right\| \leq 5\sqrt{\epsilon} + 10\sqrt{\epsilon} = 15\sqrt{\epsilon}.$$

□

Lemma 4.3.3. *For unit vector $v \in \mathbb{R}^n$ and a positive integer d, given $v^{\odot d} + E$, where E is an error vector, we can recover v'' such that for some $\alpha \in \{-1, 1\}$ we have*

$$\left\|v - \alpha v''\right\|_2 \leq \frac{2\|E\|_2}{\beta - \|E\|_2},$$

where $\beta = \frac{1}{n^{d/2-1/2}}$.

Proof. Let's for a moment work with $v^{\odot d}$ (so there is no error), and then we will take the error into account. In this case we can essentially read v off from $v^{\odot d}$. Each one-dimensional slice of $v^{\otimes d}$ (Note that as vectors, $v^{\odot d}$ and $v^{\otimes d}$ are the same; they differ only in how their entries are arranged: In the former, they are in a linear array and in the latter they are in an $n \times n \times \ldots \times n$ array. We will use them interchangeably, and we will also talk about

$v^{\otimes d} + E$ which has the obvious meaning.) is a scaled copy of v. Let us choose the copy with the maximum norm. Since $\|v\| = 1$, there is a coordinate $v(i)$ such that $|v(i)| \geq 1/\sqrt{n}$. Thus there is a one-dimensional slice of $v^{\otimes d}$ with norm at least $\frac{1}{n^{d/2-1/2}} = \beta$. Scaling this slice to norm 1 would result in αv for some $\alpha \in \{-1, 1\}$. Now, when we do have error and get $v^{\otimes d} + E$, then we must have a one-dimensional slice v' of $v^{\otimes d} + E$ with norm at least $\beta - \|E\|_2$. Then after normalizing v' to v'', one can check that $\|\alpha v'' - v\| \leq \frac{2\|E\|_2}{\beta - \|E\|_2}$ for some $\alpha \in \{-1, 1\}$. □

4.3.2 Exact analysis

We begin with the proof of the tensor decomposition theorem with access to exact tensors as stated in Theorem 4.3.1. This is essentially a structural results that says we can recover the rank-1 components when the ratios μ_i/λ_i are unique.

We first note that for a tensor T_μ with a rank-1 decomposition as in (4), that the flattened matrix version$M_\mu = \tau^{-1}(T_\mu)$ can be written as

$$M_\mu = (A^{\odot d/2})\text{diag}\,(\mu_i)\,(A^{\odot d/2})^T.$$

We will argue that the diagonalisation step works correctly (we will write $B = A^{\odot d/2}$ in what follows). The recovery of A_i from the columns of B follows by Lemma 4.3.2 above.

Our theorem is as follows (note that the first condition below is simply a normalisation of the eigenvectors):

Theorem 4.3.4. *Let $M_\mu, M_\lambda \in \mathbb{C}^{p\times p}$ such that:*

$$M_\mu = B\text{diag}\,(\mu_i)\,B^T, \qquad \textit{and} \qquad M_\lambda = B\text{diag}\,(\lambda_i)\,B^T,$$

where $B \in \mathbb{R}^{p\times m}$ and $\mu, \lambda \in \mathbb{C}^m$ for some $m \leq p$. Suppose that the following hold:

1. *For each column $B_i \in \mathbb{R}^m$ of B, $\|B_i\|_2 = 1$,*
2. *$\sigma_m(B) > 0$, and*
3. *$\mu_i, \lambda_i \neq 0$ for all i, and $\left|\frac{\mu_i}{\lambda_i} - \frac{\mu_j}{\lambda_j}\right| > 0$ for all $i \neq j$.*

*Then **Diagonalize**(M_μ, M_λ) outputs the columns of B up to sign and permutation.*

Proof. By our assumptions, the image of M_λ has dimension m and the matrix W computed in **Diagonalize**(M_μ, M_λ) satisfies $\mathsf{colspan}\,(W) = \mathsf{colspan}\,(B)$. Moreover, we could choose W to have all entries real because B is a real matrix; this will give that the ambiguities in the recovery of B are in signs and not in phase. Since the columns of W are orthonormal, the columns of $P := W^T B$ all have unit norm and it is a full rank $m \times m$ matrix. So we can write

$$W^T M_\mu W = P\mathrm{diag}\,(\mu_i)\,P^T,$$
$$(W^T M_\lambda W)^{-1} = (P^T)^{-1}\mathrm{diag}\left(\lambda_i^{-1}\right)P^{-1}.$$

Which gives

$$(W^T M_\mu W)(W^T M_\lambda W)^{-1} = P\mathrm{diag}\,(\mu_i/\lambda_i)\,P^{-1}.$$

Thus the colums of P are the eigenvectors of $(W^T M_\mu W)(W^T M_\lambda W)^{-1}$, and thus our algorithm is able to recover the columns of P up to sign and permutation. Let's call the matrix so recovered P'. Denote by $P_1, \ldots, P_m$ the columns of P, and similarly for P' and B. Then P' is given by $P'_{\pi(j)} = \alpha_j P_j$ where $\pi : [m] \to [m]$ is a permutation and $\alpha_j \in \{-1, +1\}$.

We now claim that $WP = WW^T B = B$. To see this, let $\hat{W} = [W, W']$ be an orthonormal basis that completes W. Then $\hat{W}^T\hat{W} = \hat{W}\hat{W}^T = I$. Also, $\hat{W}\hat{W}^T = WW^T + W'W'^T$. For any vector v in the span of the columns of W, we have $v = \hat{W}\hat{W}^T v = (WW^T + W'W'^T)v = WW^T v$. In other words, W acts as orthonormal matrix restricted to its image, and thus WW^T acts as the identity. In particular, $WP = WW^T B = B$.

Our algorithm has access to P' as defined above rather than to P. The algorithm will form the product WP'. But now it's clear from $WP = B$ that $WP'_{\pi(j)} = \alpha_j B_j$. Thus the algorithm will recover B up to sign and permutation. □

4.3.3 Diagonalizability and robust analysis

In applications of our tensor decomposition algorithm, we do not have access to the true underlying tensors T_μ and T_λ but rather slightly perturbed versions. We prove now that under suitably small perturbations R_μ and R_λ, we are able to recover the correct rank 1 components with good accuracy. The statement of the robust version of this theorem

closely follows that of the exact version: we merely need to add some assumptions on the magnitude of the perturbations relative to the quotients μ_i/λ_i in conditions 4 and 5.

Theorem 4.3.5. *Let $M_\mu, M_\lambda \in \mathbb{C}^{p\times p}$ such that*

$$M_\mu = B\mathrm{diag}\left(\mu_i\right)B^T, \qquad M_\lambda = B\mathrm{diag}\left(\lambda_i\right)B^T,$$

where $B \in \mathbb{R}^{p\times m}$, and $\mu, \lambda \in \mathbb{C}^m$ for some $m \leq p$. For error matrices $R_\mu, R_\lambda \in \mathbb{C}^{p\times p}$, let $M_\mu + R_\mu$ and $M_\lambda + R_\lambda$ be perturbed versions of M_μ and M_λ. Let $0 < \epsilon < 1$. Suppose that the following conditions and data are given:

1. *For each column $B_i \in \mathbb{R}^m$ of B, $\|B_i\|_2 = 1$.*
2. *$\sigma_m(B) > 0$.*
3. *$\mu_i, \lambda_i \neq 0$ for all i, $\left|\frac{\mu_i}{\lambda_i} - \frac{\mu_j}{\lambda_j}\right| \geq \Omega > 0$ for all $i \neq j$.*
4. *$0 < K_L \leq |\mu_i|, |\lambda_i| \leq K_U$.*
5. *$\|R_\mu\|_F, \|R_\lambda\|_F \leq K_1 \leq \frac{\epsilon K_L^2 \sigma_m(B)^3}{2^{11}\kappa(B)^3 K_U m^2}\min(\Omega, 1)$.*

*Then **Diagonalize** applied to $M_\mu + R_\mu$ and $M_\lambda + R_\lambda$ outputs $\tilde{B}$ such that there exists a permutation $\pi : [m] \to [m]$ and phases α_j (a phase α is a scalar in $\mathbb{C}$ with $|\alpha| = 1$) such that*

$$\left\|B_j - \alpha_j \tilde{B}_{\pi(j)}\right\| \leq \epsilon.$$

The running time of the algorithm is poly$(p, \frac{1}{\Omega}, \frac{1}{K_L}, \frac{1}{\sigma_{\min}(B)}, \frac{1}{\epsilon})$.

Proof. We begin with an informal outline of the proof. We basically implement the proof for the exact case, however because of the perturbations, various equalities now are true only approximately and this leads to somewhat lengthy and technical details, but the intuitive outline remains the same as for the exact case.

The algorithm constructs an orthonormal basis of the left singular space of $\bar{M}_\mu := M_\mu + R_\mu$; denote by Y the matrix with this basis as its columns. The fact that $\bar{M}_\mu$ is close to M_μ gives by Wedin's theorem (Theorem 3.2.3) that the left singular spaces of $\bar{M}_\mu$ and

M_μ are close. More specifically, this means that there are two matrices, W with columns forming an orthonormal basis for the left singular space of M_μ, and X with columns forming an orthonormal basis for the left singular space of $\bar{M}_\mu$ such that W and X are close in the entrywise sense. This implies that $W^T B$ and $X^T B$ are close. This can be used to show that under appropriate conditions $X^T B$ is nonsingular. Now using the fact that the columns of Y and of X span the same space, it follows that $\bar{P} := Y^T B$ is nonsingular. In the next step, we show by virtue of $\|R_\mu\|$ being small that the matrix $Y^T \bar{M}_\mu Y$ constructed by the algorithm is close to $\bar{P}\text{diag}\,(\mu_i)\,\bar{P}^T$ where the μ_i are the eigenvalues of M_μ; and similarly for $Y^T \bar{M}_\lambda Y$. We then show that $(Y^T \bar{M}_\mu Y)(Y^T \bar{M}_\lambda Y)^{-1}$ is diagonalizable and the diagonalization provides a matrix $\tilde{P}$ close to $\bar{P}$, and so $\tilde{B} = Y\tilde{P}$ gives the columns of B up to phase factors and permutation and small error.

A note on the running time. Algorithm Diagonalize uses SVD and eigenvector decomposition of diagonalizable (but not normal) matrices as subroutines. There are well-known algorithms for these as discussed earlier. The outputs of these algorithms are not exact and have a quantifiable error: The computation of SVD of $M \in \mathbb{C}^{n\times n}$ within error ϵ (for any reasonable notion of error, say $\left\|M - V\Sigma U^T\right\|_F$ where $V\Sigma U^T$ is the SVD output by the algorithm on input M) can be done in time poly $\left(n, \frac{1}{\epsilon}, \frac{1}{\sigma_{\min}(M)}\right)$. Similarly, for the eigenvector decomposition of a diagonalizable matrix $M \in \mathbb{C}^{n\times n}$ with eigenvalues $|\lambda_i - \lambda_j| \geq \Omega > 0$ for $i \neq j$, we can compute the decomposition within error ϵ in time poly$(n, \frac{1}{\Omega}, \frac{1}{\epsilon}, \frac{1}{\min_i |\lambda_i|})$.

In the analysis below, we ignore the errors from these computations as they can be controlled and will be of smaller order than the error from the main analysis. This can be made rigorous but we omit the details in the interest of brevity. Combining the running time of the two subroutines one can check easily that the overall running time is what is claimed in the statement of the theorem.

We now proceed with the formal proof. The proof is broken into 7 steps.

Step 1. $W^T B \approx X^T B$.

Let $\bar{M}_\mu := M_\mu + R_\mu$ and $\bar{M}_\lambda := M_\lambda + R_\lambda$. Now the fact that $\|R_\mu\|_F$ is small implies

by Wedin's theorem (Theorem 3.2.3) that the left singular spaces of M_μ and $\bar{M}_\mu$ are close: Specifically, by Theorem IV.1.8 in [26] about canonical angles between subspaces, we have: There exists an orthonormal basis of the left singular space of M_μ (given by the columns $w_1, \dots w_m$ of $W \in \mathbb{C}^{p\times m}$) and an orthonormal basis of the left singular space of $\bar{M}_\mu$ (given by the columns $x_1, \dots, x_m$ of $X \in \mathbb{C}^{p\times m}$) such that

$$x_j = c_j w_j + s_j z_j, \text{ for all } j,$$

where $0 \le c_1 \le \ldots \le c_m \le 1$, and $1 \ge s_1 \ge \ldots \ge s_m \ge 0$, and $c_j^2 + s_j^2 = 1$ for all j; vectors $w_1, \dots, w_m; z_1, \dots, z_m$ form an orthonormal basis. (For the last condition to hold we need $p \ge 2m$. A similar representation can be derived when this condition does not hold and the following computation will still be valid. We omit full discussion of this other case for brevity; in any case, we could arrange so that $p \ge 2m$ without any great penalty in the parameters achieved.) We now apply Wedin's theorem 3.2.3 to M_μ and $\bar{M}_\mu$ to upper bound s_j. To this end, first note that by Claim 3.2.1 we have $\sigma_m(M_\mu) \ge K_L \sigma_m(B)^2$; and second, by Weyl's inequality for singular values (Lemma 3.2.4) we have $\left|\sigma_j(\bar{M}_\mu) - \sigma_j(M_\mu)\right| \le \sigma_1(R_\mu) \le K_1$ for all j. Thus in Theorem 3.2.3, with Σ_1 corresponding to non-zero singular values of M_μ, we have $\max \sigma(\Sigma_2) = 0$. And we can choose a corresponding conformal SVD of $\bar{M}_\mu$ so that $\min \sigma(\bar{\Sigma}_1) \ge K_L \sigma_m(B)^2 - K_1$. Which gives, $\|\sin \Phi\|_2 \le K_1/(K_L \sigma_m(B)^2 - K_1) =: K_2$, where Φ is the matrix of canonical angles between $\mathsf{colspan}\,(W)$ and $\mathsf{colspan}\,(X)$. Thus we have

$$s_j \le K_2, \tag{7}$$

for all j.

Now we can show that $X^T B$ is close to $W^T B$: The (i,j)'th entry of $W^T B - X^T B$ is $(1 - c_i) w_i^T b_j - s_i z_i^T b_j$. Using (7) and $\|w_i\|, \|b_j\|, \|z_i\| \le 1$, we have

$$(1 - c_i) w_i^T b_j - s_i z_i^T b_j \le s_i^2 + s_i \le 2K_2.$$

And so $\left\|W^T B - X^T B\right\|_F \le 2m^2 K_2$. Hence by Lemma 3.2.4 we have $\left|\sigma_j(W^T B) - \sigma_j(X^T B)\right| \le 2m^2 K_2$ for all j.

Step 2. $\bar{P} := Y^T B$ *is full rank.*

The singular values of $W^T B$ are the same as those of B. Briefly, this is because W^T acts as an isometry on $\mathsf{colspan}\,(B)$. Also observe that the singular values of $Y^T B$ are the same as those of $X^T B$. Briefly, this is because Y^T and X^T act as isometries on $\mathsf{colspan}\,(X) = \mathsf{colspan}\,(Y)$. These two facts together with the closeness of the singular values of $W^T B$ and $X^T B$ as just shown imply that

$$\left|\sigma_j(B) - \sigma_j(Y^T B)\right| \leq 2m^2 K_2 \tag{8}$$

for all j. Now using that $2m^2 K_2 < \sigma_m(B)/2$ (This follows by our condition 5 in the theorem giving an upper bound on K_1: $K_1 \leq \epsilon \frac{K_L}{K_U} \frac{K_L \sigma_m(B)^3}{2^{11}\kappa(B)^3 m^2}$ which gives $K_1 \leq \frac{K_L \sigma_m(B)^3}{8m^2}$. This in turn implies $2m^2 K_2 < \sigma_m(B)/2$ using $\sigma_m(B) \leq 1$; we omit easy verification.) we get that $\sigma_m(Y^T B) > 0$ and hence $Y^T B$ is full rank. We note some consequences of (8) for use in later steps:

$$\kappa(\bar{P}) \leq 4\kappa(B). \tag{9}$$

This follows from $\kappa(\bar{P}) \leq \frac{\sigma_1(B)+2m^2K_2}{\sigma_m(B)-2m^2K_2} \leq 4\kappa(B)$, because $2m^2 K_2 < \sigma_m(B)/2$.

$$\sigma_m(\bar{P}) \leq \sigma_m(B) + 2m^2 K_2 < 2\sigma_m(B). \tag{10}$$

$$\sigma_m(\bar{P}) \geq \sigma_m(B) - 2m^2 K_2 < \sigma_m(B)/2. \tag{11}$$

Step 3. $Y^T \bar{M}_\mu Y \approx \bar{P}\mathrm{diag}\,(\mu_i)\,\bar{P}^T$ *and* $Y^T \bar{M}_\lambda Y \approx \bar{P}\mathrm{diag}\,(\lambda_i)\,\bar{P}^T$.

More precisely, let $E_\mu := Y^T \bar{M}_\mu Y - \bar{P}\mathrm{diag}\,(\mu_i)\,\bar{P}^T$, then $\|E_\mu\|_F \leq m^2 \|R_\mu\|_F$; and similarly for $\bar{M}_\lambda, E_\lambda := Y^T \bar{M}_\lambda Y - \bar{P}\mathrm{diag}\,(\lambda_i)\,\bar{P}^T$. The proof is short: We have $Y^T \bar{M}_\mu Y = Y^T(M_\mu + R_\mu)Y = Y^T M_\mu Y + Y^T R_\mu Y = \bar{P}\mathrm{diag}\,(\mu_i)\,\bar{P}^T + Y^T R_\mu Y$. Hence $\|E_\mu\|_F = \left\|Y^T R_\mu Y\right\|_F \leq \|R_\mu\|_F$.

Step 4. $(Y^T \bar{M}_\mu Y)(Y^T \bar{M}_\lambda Y)^{-1}$ *is diagonalizable.*

This is because Theorem 4.3.6 is applicable to $\tilde{N} := (Y^T \bar{M}_\mu Y)(Y^T \bar{M}_\lambda Y)^{-1} = (\bar{P}\mathrm{diag}\,(\mu_i)\,\bar{P}^T + E_\mu)(\bar{P}\mathrm{diag}\,(\lambda_i)\,\bar{P}^T + E_\lambda)^{-1}$: using $\|E_\mu\|_F \leq \|R_\mu\|_F$, the two condition to verify are

- $\frac{6\kappa(\bar{P})^3 m K_U}{K_L^2 \sigma_m(\bar{P})^2} K_1 \leq \Omega$.

  This follows from our condition 5 using (9), (11) and $\sigma_m(B) \leq 1$.

- $K_1 \leq \sigma_m(\bar{P})^2 K_L/2$.

  This also follows from condition 5, using (10) and $\epsilon \leq 1$.

Hence $\tilde{N}$ is diagonalizable: $\tilde{N} = \tilde{P}\mathrm{diag}\,(\tilde{\gamma}_i)\,\tilde{P}^{-1}$.

Step 5. *The eigenvalues of $\tilde{N}$ are close to the eigenvalues of $\bar{P}\mathrm{diag}\,(\mu_i/\lambda_i)\,\bar{P}^T$.* This follows from our application of Theorem 4.3.6 in the previous step (specifically from (16)) and gives a permutation $\pi : [m] \to [m]$ such that

$$\left|\frac{\mu_i}{\lambda_i} - \tilde{\gamma}_{\pi(i)}\right| < \Omega/2,$$

where the $\tilde{\gamma}_i$ are the eigenvalues of $\tilde{N}$.

In the next step we show that there exist phases α_i such that $\tilde{P}^{\pi,\alpha} := [\alpha_1 \tilde{P}_{\pi(1)}, \alpha_2 \tilde{P}_{\pi(2)}, \ldots, \alpha_m \tilde{P}_{\pi(m)}]$ is close to $\bar{P}$.

Step 6. *$\bar{P}$ is close to $\tilde{P}$ up to sign and permutation of columns.*
We upper bound the angle θ between the corresponding eigenpairs $(\frac{\mu_j}{\lambda_j}, \bar{P}_j)$ and $(\tilde{\gamma}_{\pi(j)}, \tilde{P}_{\pi(j)})$ of $N := \bar{P}\mathrm{diag}\,(\mu_i/\lambda_i)\,\bar{P}^{-1}$ and $\tilde{N}$. Theorem 3.2.9 (a generalized version of the $\sin(\theta)$ eigenspace perturbation theorem for diagonalizable matrices) applied to N and $\tilde{N}$ gives (with the notation derived from Theorem 3.2.9)

$$\sin\theta \leq \kappa(Z_2)\frac{\left\|(N - \tilde{\gamma}_{\pi(j)} I)\tilde{P}_{\pi(j)}\right\|_2}{\min_i \left|(N_2)_{ii} - \tilde{\gamma}_{\pi(j)}\right|}.$$

To bound the RHS above, we will estimate each of the three terms. The first term:

$$\kappa(Z_2) \leq \kappa(\bar{P}^{-1}) = \kappa(\bar{P}) \leq 4\kappa(B),$$

where for the first inequality we used that the condition number of a submatrix can only be smaller [124]; the second inequality is (9).

Setting $\mathsf{Err} := N - \tilde{N}$, we bound the second term:

$$\begin{aligned}
\left\|(N - \tilde{\gamma}_{\pi(j)} I)\tilde{P}_{\pi(j)}\right\|_2 &= \left\|(\tilde{N} - \tilde{\gamma}_{\pi(j)} I)\tilde{P}_{\pi(j)} + \mathsf{Err}\ \tilde{P}_{\pi(j)}\right\|_2 \\
&= \left\|\mathsf{Err}\ \tilde{P}_{\pi(j)}\right\|_2 \\
&\leq \|\mathsf{Err}\|_2 \\
&\leq \kappa(\bar{P})^2 \cdot \frac{K_U}{K_L} \cdot 2m \cdot \frac{K_1}{\sigma_m(\bar{P})^2 K_L} \quad \text{(by (15) in Theorem 4.3.6)} \\
&\leq 2^6 \kappa(B)^2 m \frac{K_U}{K_L} \frac{K_1}{\sigma_m(B)^2 K_L} \quad \text{(using (9), (10)).} \qquad (12)
\end{aligned}$$

And lastly, the third term:

$$\begin{aligned}
\min_i \left|(N_2)_{ii} - \tilde{\gamma}_{\pi(j)}\right| &\geq \min_{i:i\neq j} \left|\frac{\mu_i}{\lambda_i} - \frac{\mu_j}{\lambda_j}\right| - \left|\frac{\mu_j}{\lambda_j} - \tilde{\gamma}_{\pi(j)}\right| \\
&\geq \Omega - \kappa(\bar{P}) \|\mathsf{Err}\|_2 \quad \text{(using Lemma 3.2.7)} \\
&\geq \Omega - 2^9 \kappa(B)^3 m \frac{K_U}{K_L} \frac{K_1}{\sigma_m(B)^2 K_L} \quad \text{(using (12) and (9)).}
\end{aligned}$$

To abbreviate things a bit, let's set $\epsilon' := 2^9 \kappa(B)^3 \frac{K_U}{K_L} m \frac{K_1}{\sigma_m(B)^2 K_L}$. Then, putting things together we get

$$\sin(\theta) \leq \frac{\epsilon'}{\Omega - \epsilon'}.$$

Now using the fact that the columns of $\tilde{P}$ and $\bar{P}$ are unit length implies that there exist phases α_i such that

$$\left\|\alpha_j \tilde{P}_{\pi(j)} - \bar{P}_j\right\|_2 \leq \frac{\epsilon'}{\Omega - \epsilon'}. \qquad (13)$$

Step 7. *$Y\tilde{P}$ gives B approximately and up to phase factors and permutation of columns.* This follows from two facts: (1) $\tilde{P}^{\pi,\alpha} \approx \bar{P}$, so $Y\tilde{P}^{\pi,\alpha} \approx Y\bar{P}$ (we will prove this shortly); and (2) $Y\bar{P} = YY^T B$ (follows by the definition of $\bar{P}$). Now note that the operator YY^T is projection to $\mathsf{colspan}\,(Y)$; since the angle between $\mathsf{colspan}\,(Y)$ and $\mathsf{colspan}\,(B)$ is small as we showed in Step 1, we get that $YY^T B \approx B$.

Formally, we have

$$\left\|Y\alpha_j \tilde{P}_{\pi(j)} - Y\bar{P}_j\right\|_2 \leq \|Y\|_2 \left\|\alpha_j \tilde{P}_{\pi(j)} - \bar{P}_j\right\|_2 \leq \frac{\epsilon'}{\Omega - \epsilon'},$$

using (13). And

$$\left\|b_j - YY^T b_j\right\|_2 \leq K_2,$$

where the last inequality used that the sine of the angle between $\mathsf{colspan}\,(Y)$ and $\mathsf{colspan}\,(W) = \mathsf{colspan}\,(B)$ is at most K_2 as proved in Step 1.

Putting these together we get

$$\left\|Y\alpha_j\tilde{P}_{\pi(j)} - b_j\right\|_2 \leq \left\|b_j - YY^T b_j\right\|_2 + \left\|Y\alpha_j\tilde{P}_{\pi(j)} - Y\bar{P}_j\right\|_2 \leq \frac{\epsilon'}{\Omega - \epsilon'} + K_2.$$

Letting $\tilde{B} = Y\tilde{P}$ gives

$$\left\|\alpha_j\tilde{B}_{\pi(j)} - b_j\right\|_2 \leq \frac{\epsilon'}{\Omega - \epsilon'} + K_2 \leq \epsilon.$$

The last inequality follows from our condition 5, which implies that $\frac{\epsilon'}{\Omega-\epsilon'} \leq \epsilon/2$ and $K_2 \leq \epsilon/2$.

□

Theorem 4.3.6 (Diagonalizability of perturbed matrices)**.** *Let $N_\mu, N_\lambda \in \mathbb{C}^{m\times m}$ be full rank complex matrices such that $N_\mu = Q\mathrm{diag}\,(\mu_i)\,Q^T, N_\lambda = Q\mathrm{diag}\,(\lambda_i)\,Q^T$ for some $Q \in \mathbb{R}^{m\times m}$ and $\mu, \lambda \in \mathbb{C}^m$. Suppose we also have the following conditions and data:*

1. $0 < K_L \leq |\mu_i|, |\lambda_i| \leq K_U$.
2. $|\mu_i/\lambda_i - \mu_j/\lambda_j| > \Omega > 0$ *for all pairs* $i \neq j$.
3. $0 < K < 1$ *and* $E_\mu, E_\lambda \in \mathbb{C}^{m\times m}$ *such that* $\|E_\mu\|_F, \|E_\lambda\|_F \leq K$.
4. $6\kappa(Q)^3 \cdot \frac{K_U}{K_L} \cdot m \cdot \frac{K}{\sigma_m(Q)^2 K_L} \leq \Omega$.
5. $K \leq \sigma_m(Q)^2 K_L/2$.

Then $(N_\mu + E_\mu)(N_\lambda + E_\lambda)^{-1}$ is diagonalizable and hence has n eigenvectors.

Proof. Defining $F_\mu := (Q\mathrm{diag}\,(\mu_i)\,Q^T)^{-1}E_\mu$, and similarly F_λ, we have

$$
\begin{aligned}
(N_\mu + E_\mu)(N_\lambda + E_\lambda)^{-1} &= (Q\mathrm{diag}\,(\mu_i)\,Q^T + E_\mu)(Q\mathrm{diag}\,(\lambda_i)\,Q^T + E_\lambda)^{-1} \\
&= Q\mathrm{diag}\,(\mu_i)\,Q^T(I + F_\mu)(I + F_\lambda)^{-1}(Q\mathrm{diag}\,(\lambda_i)\,Q^T)^{-1} \\
&= Q\mathrm{diag}\,(\mu_i)\,Q^T(I + F_\mu)(I + G_\lambda)(Q\mathrm{diag}\,(\lambda_i)\,Q^T)^{-1} \qquad (14) \\
&= Q\mathrm{diag}\,(\mu_i/\lambda_i)\,Q^{-1} + \mathsf{Err}.
\end{aligned}
$$

In (14) above $G_\lambda = (I + F_\lambda)^{-1} - I$; hence by Claim 3.2.2 (which is applicable because $\|F_\lambda\|_F \leq \frac{K}{\sigma_m(Q)^2 K_L} \leq 1/2$, by our assumption) we have $\|G_\lambda\|_F \leq (m+1)\,\|F_\lambda\|_F$. The norm of Err then satisfies

$$
\begin{aligned}
\|\mathsf{Err}\|_F &\leq \frac{\sigma_1(Q)^2}{\sigma_m(Q)^2} \cdot \frac{K_U}{K_L}\left(\|F_\mu\|_F + (m+1)\,\|F_\lambda\|_F + (m+1)\,\|F_\mu\|_F \cdot \|F_\lambda\|_F\right) \\
&\leq \kappa(Q)^2 \cdot \frac{K_U}{K_L} \cdot 2m \cdot \frac{K}{\sigma_m(Q)^2 K_L}. \qquad (15)
\end{aligned}
$$

Now note that $3\kappa(Q)\,\|\mathsf{Err}\|_2 \leq 6\kappa(Q)^3 \cdot \frac{K_U}{K_L} \cdot m \cdot \frac{K}{\sigma_m(Q)^2 K_L} \leq \Omega$ by our assumption and so Lemma 3.2.7 is applicable with matrices $Q\mathrm{diag}\,(\mu_i/\lambda_i)\,Q^{-1}$, Q, and Err playing the roles of A, X, and E, resp. Lemma 3.2.7 gives us a permutation $\pi : [m] \to [m]$ such that

$$
\left|\nu_{\pi(i)}(Q\mathrm{diag}\,(\mu_i/\lambda_i)\,Q^{-1} + \mathsf{Err}) - \nu_i(Q\mathrm{diag}\,(\mu_i/\lambda_i)\,Q^{-1})\right| \leq \kappa(Q)\,\|\mathsf{Err}\|_2 < \Omega/2, \qquad (16)
$$

where $\nu_i(M)$ denotes an eigenvalue of M.

Hence all the eigenvalues of $(N_\mu + E_\mu)(N_\lambda + E_\lambda)^{-1}$ are distinct. By Lemma 3.2.8, it has n linearly independent eigenvectors $\{v_1, \ldots, v_n\}$. □

4.3.4 Genericity of mixing matrices and average case analysis

Our necessary condition for identifiability is satisfied almost surely by randomly chosen vectors for a fairly general class of distributions. For simplicity we restrict ourselves to the case of $d = 2$ and Gaussian distribution in the following theorem; the proof of a more general statement would be similar.

Theorem 4.3.7. *Let $v_1, \ldots, v_m \in \mathbb{R}^n$ be standard Gaussian i.i.d. random vectors, with $m \leq \binom{n+1}{2}$. Then $v_1^{\otimes 2}, \ldots, v_m^{\otimes 2}$ are linearly independent almost surely.*

Proof Sketch. Let's take $m = \binom{n+1}{2}$ without loss of generality. Consider vextors $w_1, \ldots, w_m$, where w_i is obtained from $v_i^{\otimes 2}$ by removing duplicate components; e.g., for $v_1 \in \mathbb{R}^2$, we have $v_1^{\oplus 2} = (v_1(1)^2, v_1(2)^2, v_1(1)v_1(2), v_2(1)v_1(2))$ and $w_1 = (v_1(1)^2, v_1(2)^2, v_2(1)v_1(2))$. Thus $v_i \in \mathbb{R}^{\binom{n+1}{2}}$. Now consider the determinant of the $\binom{n+1}{2} \times \binom{n+1}{2}$ matrix with the w_i as columns. As a formal multivariate polynomial with the components of the v_i as variables, this determinant is not identically 0. This is because, for example, it can be checked that the monomial $w_1(1)^2 \ldots w_n(n)^2 w_{n+1}(\rho(n+1)) \ldots w_m(\rho(m))$ occurs precisely once in the expansion of the determinant as a sum of monomials (here $\rho : \{n+1, \ldots, m\} \to \binom{[n]}{2}$ is an arbitrary bijection). The proof can now be completed along the lines of the well-known Schwartz–Zippel lemma. □

We now show that the condition number of the Khatri–Rao power of a random matrix behaves well in certain situations. For simplicity we will deal with the case where the entries of the base matrix M are chosen from $\{-1, 1\}$ uniformly at random; the case of Gaussian entries also gives a similar though slightly weaker result, but would require some extra work.

We define a notion of d'th power of a matrix $M \in \mathbb{R}^{n \times m}$ which is similar to the Khatri–Rao power except that we only keep the non-redundant multilinear part resulting in $\binom{n}{d} \times m$ matrix. Working with this multilinear part will simplify things. Formally, $M^{\ominus d} := [M_1^{\ominus d}, \ldots, M_m^{\ominus d}]$, where for a column vector $C \in \mathbb{R}^n$, define $C^{\ominus d} \in \mathbb{R}^{\binom{n}{d}}$ with entries given by $C_S := C_{i_1} C_{i_2} \ldots C_{i_d}$ where where $1 \le i_1 < i_2 < \ldots < i_d \le n$ and $S = \{i_1, \ldots, i_d\} \in \binom{[n]}{d}$.

The following theorem is stated for the case when the base matrix $M \in \mathbb{R}^{n \times n^2}$. This choice is to keep the statement and proof of the theorem simple; generalization to more general parameterization is straightforward. While the theorem below is proved for submatrices $M^{\ominus d}$ of the Khatri–Rao power $M^{\odot d}$, similar results hold for $M^{\odot d}$ by the interlacing properties of the singular values of submatrices [124].

Theorem 4.3.8. *Let $M \in \mathbb{R}^{n \times m}$ be chosen by sampling each entry iid uniformly at random from $\{-1, 1\}$. For $m = n^2$, integer $d \ge 3$, and $N = \binom{n}{d}$, and $A = M^{\ominus d}$ we have*

$$\mathbb{E} \max_{j \le n^2} \left| \sigma_j(A) - \sqrt{N} \right| < N^{1/2 - \Omega(1)}.$$

Proof. We are going to use Theorem 5.62 of Vershynin [130] which we state here essentially verbatim:

Theorem 4.3.9 ([130])**.** *Let A be an $N \times m$ matrix ($N \geq m$) whose columns A_j are independent isotropic random vectors in $\mathbb{R}^N$ with $\|A_j\|_2 = \sqrt{N}$ almost surely. Consider the incoherence parameter*

$$\mu := \frac{1}{N}\mathbb{E}\max_{j\leq m}\sum_{k\in[m],k\neq j}\langle A_j, A_k\rangle^2 .$$

Then for absolute constants C, C_0 we have $\mathbb{E}\left\|\frac{1}{N}A^*A - I\right\| \leq C_0\sqrt{\frac{\mu\log m}{N}}$. In particular,

$$\mathbb{E}\max_{j\leq m}\left|\sigma_j(A) - \sqrt{N}\right| < C\sqrt{\mu\log m}.$$

Our matrix $A = M^{\ominus d}$ will play the role of matrix A in Theorem 4.3.9. Note that for a column A_j we have $\mathbb{E}A_j \otimes A_j = I$, so the A_j are isotropic. Also note that $\|A_j\|_2 = \sqrt{N}$ always.

We now bound the incoherence parameter μ. To this end, we first prove a concentration bound for $\langle A_j, A_k\rangle$, for fixed j, k. We use a concentration inequality for polynomials of random variables. Specifically, we use Theorem 23 at (http://www.contrib.andrew.cmu.edu/~ryanod/?p=1472 Let us restate that theorem here.

Theorem 4.3.10. *Let $f : \{-1,1\}^n \to \mathbb{R}$ be a polynomial of degree at most k. Then for any $t \geq (2e)^{k/2}$ we have*

$$\Pr_{x\sim\{-1,1\}}[|f(x)| \geq t\|f\|_2] \leq \exp\left(-\frac{k}{2e}t^{2/k}\right).$$

Here $\|f\|_2 := [\mathbb{E}_x f(x)^2]^{1/2}$. For our application to $\langle A_j, A_k\rangle$, we first fix A_j arbitrarily. Then $\langle A_j, A_k\rangle$, which will play the role of $f(x)$ in the above theorem, can be written as $\sum_{S\in\binom{[n]}{d}} c_S x_S$ where the choice of the coefficients $c_S = \pm 1$ comes from the fixing of A_j and the entries of A_k are of the form x_S, where $S \in \binom{[n]}{d}$. Now

$$\begin{aligned}
\mathbb{E}_x \langle A_j, A_k \rangle^2 &= \sum_{S,S' \in \binom{[n]}{d}} c_S c_{s'} \mathbb{E}_x x_S x_{S'} \\
&= \sum_{S \in \binom{[n]}{d}} c_S^2 \mathbb{E}_x x_S^2 + \sum_{S,S' \in \binom{[n]}{d} : S \neq S'} c_S c_{S'} \mathbb{E}_x x_S x_{S'} \\
&= \binom{n}{d} \\
&= N.
\end{aligned}$$

In other words, for our choice of f we have $\|f\|_2 = \sqrt{N}$.

Applying Theorem 4.3.10 with $t \geq (2e)^{d/2}$ and $\lambda = t\sqrt{N}$ we have

$$\Pr_{x \sim \{-1,1\}}[|\langle A_j, A_k \rangle| \geq \lambda] \leq \exp\left(-\frac{d}{2e} t^{2/d}\right) = \exp\left(-\frac{d}{2e} \frac{\lambda^{2/d}}{N^{1/d}}\right). \tag{17}$$

Note that we proved the above inequality for any fixed A_j, so clearly it also follows when A_j is also random.

We now estimate parameter μ. Note that $\langle A_j, A_k \rangle^2 \leq N^2$ always. When the union of the event in (17) over all $j \neq k$, which we denote by B, does not hold, we will use the bound just mentioned. For the following computation recall that the number of columns m in A is n^2.

$$\begin{aligned}
\mu &\leq \frac{1}{N} m\lambda^2 \mathsf{Pr}\left(\bar{B}\right) + \frac{1}{N} m N^2 \mathsf{Pr}\left(B\right) \\
&\leq \frac{m\lambda^2}{N} + \frac{m\binom{m}{2} N^2}{N} \exp\left(-\frac{d}{2e} \frac{\lambda^{2/d}}{N^{1/d}}\right) \\
&\leq \frac{n^2\lambda^2}{N} + n^6 N \exp\left(-\frac{d}{2e} \frac{\lambda^{2/d}}{N^{1/d}}\right).
\end{aligned} \tag{18}$$

Now choose $\lambda := N^{1/2+\epsilon}$ for a small $\epsilon > 0$. Then the expression in (18) is bounded by

$$(18) \leq n^2 N^{2\epsilon} + n^6 N \exp\left(-\frac{d}{2e} N^{2\epsilon/d}\right).$$

It's now clear that for a sufficiently small choice of ϵ (say 0.05) and sufficiently large n (cepending on d and ϵ), only the first term above is significant and using our assumption $d > 2$ gives

$$\mu < 2n^2 N^{2\epsilon} < 2d! N^{2/d+\epsilon} << N.$$

Therefore by Theorem 4.3.9 we have

$$\mathbb{E}\left\|\frac{1}{N}A^*A - I\right\| \leq C_0\sqrt{\frac{\mu \log n^2}{N}} < 1/N^{\Omega(1)},$$

which gives

$$\mathbb{E}\max_{j\leq n^2}\left|\sigma_j(A) - \sqrt{N}\right| < N^{1/2-\Omega(1)}.$$

□

In particular, setting $s_{\min}(A) := s_{n^2}(A)$ we have

$$\mathbb{E}\left|\sigma_{\min}(A) - \sqrt{N}\right| < 1/N^{1/2-\Omega(1)}.$$

Using Markov this also gives probability bounds.

CHAPTER V

ALGORITHMS FOR UNSUPERVISED LEARNING

5.1 Introduction

Having developed the tensor machinery in the previous chapter, we are now ready to tackle the unsupervised learning problems posed in Chapter 2. We will first apply the additive subspace tensor decomposition (Section 4.2) to the subspace junta problem (Problem 5). Next, we shall apply our robust pairwise tensor decomposition to fully-determined ICA (the case when $m = n$ and the mixing matrix is square in Problem 1), Gaussian mixtures (Problem 3), and underdetermined ICA (Problem 2), the case when $m > n$). Our focus will be on quantitatively efficient algorithms, that is, algorithms that run in polynomial time (in terms of the dimensionality of the space and error parameters), succeed with high probability, and require only a polynomial number of samples. We will now briefly survey the major unsupervised learning results of this chapter.

The additive subspace tensor decomposition provides an approach to the problem of factoring distributions (Problem 5), learning a k-subspace junta (Problem 6), and in particular an algorithm for the Gaussian noise model (Problem 7). In particular, we shall show that factorizable distributions (as in Problem 5) have moment tensors which have additive decompositions. This key property is proved in a structural lemma, Lemma 5.2.5, and is inspired by [117, 55]. Our lemma is a substantial generalization of those earlier results. We then carefully apply our methods under the assumption that the distribution over the irrelevant subspace follows the Gaussian noise model of Problem 7. Recall that the idea behind this assumption is that the irrelevant subspace is entirely separate from the features of interest, and that they are essentially measurements of some other unrelated quantity which is subject to Gaussian measurement error. Our framework for k-subspace juntas is quite general, and to develop specific applications, we are obliged to further refine our models. Thus, much of Section 5.2 is devoted to formulating the correct (geometric) model for

learning in this setting. Nonetheless, we are able to give a number of interesting examples in Section 5.2.5 with very strong complexity guarantees.

We are able to productively handle a broad swathe of questions using the rank 1 decomposition algorithm. We will review the highlights here briefly. The first application of our rank 1 tensor decomposition in Section 4.3 is to the fully determined ICA model. To apply the tensor decomposition, we have to very carefully pick two tensors as input – these turn out to be the Hessian derivative matrix of suitable functions of the Fourier transform of the distribution evaluated at random points (recall that the derivatives of a multivariate function constitute a tensor field, so by evaluating the derivative at different points, we get different tensors). Fortuitously, these derivatives all share the same rank 1 components, and we are able to apply our tensor decomposition algorithm to recover the columns of the mixing matrix; what's happening in the background is some magic with a multi-variate higher order differentiation chain rule. We call our general technique *Fourier PCA*. This is motivated by the fact that in the base case of second derivatives, our algorithms look very much like PCA of a reweighted covariance matrix, where each data pointed is weighted by a complex exponential. We are able to give an alternative, very efficient (both in theory and practice) algorithm by extending these techniques via a polynomial gap concentration inequality. A second exension is to removing certain types of measurement noise – Gaussian functions hold a special place in harmonic analysis because they interact very well with Fourier transforms; using such ideas, we can extend our algorithm to handle unknown Gaussian noise. The ICA model with Gaussian noise is given by

$$x = As + \eta,$$

where $\eta \sim N(0, \Sigma)$ is independent Gaussian noise with unknown general covariance matrix $\Sigma \in \mathbb{R}^{n \times n}$. Thus we are able to solve ICA (both fully determined and underdetermined) in the presence of Gaussian noise. None of our ICA results require full independence of the s_i, it is sufficient to have d-wise independence.

Our Gaussian mixture model result applies the same method to learning mixtures of spherical Gaussians (see the full version). Using Fourier PCA, we recover the result of Hsu

and Kakade [73], and extend it to the setting of noisy mixtures, where the noise itself is an unknown arbitrary Gaussian. Our result can be viewed as saying that reweighted PCA gives an alternative algorithm for learning such mixtures.

Theorem 5.1.1. *Fourier PCA for Mixtures applied to a mixture of $k < n$ spherical Gaussians $N(\mu_i, \sigma_i^2 I_n)$ recovers the parameters of the mixture to desired accuracy ϵ using time and samples polynomial in $k, n, 1/\epsilon$ with high probability, assuming that the means μ_i are linearly independent.*

Note that this method is still a matrix method, and is only capable of dealing with $k < n$ Gaussians. Subsequent authors have extended this work, using our underdetermined ICA algorithm and are capable of separating mixtures of many more Gaussians [13].

None of the above results, however, uses the full power of our tensor decomposition – our underdetermined ICA algorithm does. We rely here on the higher order derivatives of the log of the Fourier transform of the distribution, once again evaluated at random Fourier coefficients. The input to the algorithms, apart from the samples generated according to the unknown noisy underdetermined ICA model, consists of several parameters whose meaning will be clear in the theorem statement below: a tensor order parameter d, number of signals m, accuracy parameter ϵ, confidence parameter δ, bounds on moments and cumulants M and Δ, an estimate of the conditioning parameter σ_m, and moment order k. The notation $A^{\odot d}$ used below is explained in the preliminaries section; briefly, it's a $n^d \times m$ matrix with each column obtained by flattening $A_i^{\otimes d}$ into a vector.

Theorem 5.1.2. *Let $x \in \mathbb{R}^n$ be given by an underdetermined ICA model with unknown Gaussian noise $x = As + \eta$ where $A \in \mathbb{R}^{n \times m}$ with unit norm columns and the covariance matrix $\Sigma \in \mathbb{R}^{n \times n}$ are unknown. Let $d \in 2\mathbb{N}$ be such that $\sigma_m(A^{\odot d/2}) > 0$. Let M_k, M_d, M_{2d} and $k > d$ be such that for each s_i, there is a k_i satisfying $d < k_i < k$ and $|\mathrm{cum}_{k_i}(s_i)| \geq \Delta$, and $\mathbb{E}\left(|s_i|^{k_i}\right), \mathbb{E}\left(\sigma_1(\Sigma)^k\right) \leq M_k$, $\mathbb{E}\left(|s_i|^d\right) \leq M_d$, and $\mathbb{E}\left(|s_i|^{2d}\right) \leq M_{2d}$. Then one can recover the columns of A up to ϵ accuracy in 2-norm and up to the sign using poly$\left(m^k, M_d^k, M_{2d}, 1/\Delta, 1/\sigma_m(A^{\odot d/2})^k, 1/\epsilon, 1/\sigma^k\right)$ samples and with similar polynomial time complexity with probability at least $3/4$, where $0 < \sigma < \frac{\Delta}{M_k} poly(\sigma_m(m^k, A^{\odot d/2})^k, 1/k^k)$.*

The probability of success of the algorithm can be boosted from $3/4$ to $1-\delta$ for any $\delta > 0$ by taking $O(\log(1/\delta))$ independent runs of the algorithm and using an adaptation of the "median" trick (see e.g., Thm 2.8 in[97]). To our knowledge, this is the first polynomial-time algorithm for underdetermined ICA with provable finite sample guarantees. It works under mild assumptions on the input distribution and nondegeneracy assumptions on the mixing matrix A. The assumption says that the columns of the matrix when tensored up individually are linearly independent. For example, with $d = 4$, suppose that every s_i differs from a Gaussian in the fifth or higher moment by Δ, then we can recover all the components as long as $\text{vec}\left(A_i A_i^T\right)$ are linearly independent. Thus, the number of components that can be recovered can be as high as $m = n(n+1)/2$. Clearly, this is a weakening of the standard assumption that the columns of A are linearly independent. This assumption can be regarded as a certain incoherence type assumption. Moreover, in a sense it's a necessary and sufficient condition: the ICA problem is solvable for matrix A if and only if any two columns are linear independent [47], and this turns out to be equivalent to the existence of a finite d such that $A^{\odot d}$ has full column rank. A well-known condition in the literature on tensor decomposition is Kruskal's condition [89]. Unlike that condition it is easy to check if a matrix satisfies our assumption (for a fixed d). Our assumption is true *generically*: For a randomly chosen matrix $A \in \mathbb{R}^{n \times \binom{n}{d}}$ (e.g. each entry being i.i.d. standard Gaussian), we have $\sigma_{min}(A^{\odot d}) > 0$ with probability 1. In a similar vein, for a randomly chosen matrix $A \in \mathbb{R}^{n \times \binom{n}{d}}$ its condition number is close to 1 with high probability; see Theorem 4.3.8 for a precise statement and proof. Moreover, our assumption is robust also in the smoothed sense [13]: if we start with an arbitrary matrix $M \in \mathbb{R}^{n \times \binom{n}{2}}$ and perturb it with a noise matrix $N \in \mathbb{R}^{n \times \binom{n}{2}}$ with each entry independently chosen from $N(0, \sigma^2)$, then we have $\sigma_{min}((M+N)^{\odot 2}) = \sigma^2/n^{O(1)}$ with probability at least $1 - 1/n^{\Omega(1)}$, and a similar generalization holds for higher powers. This follows from a simple application of the anti-concentration properties of polynomials in independent random variables; see [13] for a proof. See also [25].

As in the fully-determined ICA setting, we require that our random variables have some cumulant different from a Gaussian. One aspect of our result is that using the d^{th} derivative,

one loses the ability to detect non-Gaussian cumulants at order d and lower; on the other hand, a theorem of Marcinkiewicz [98] implies that this does not cause a problem.

Theorem 5.1.3 (Marcinkiewicz)**.** *Suppose that random variable $x \in \mathbb{R}$ has only a finite number of non-zero cumulants, then x is distributed according to a Gaussian, and every cumulant of order greater than 2 vanishes.*

Thus, even if we miss the difference in cumulants at order $i \leq d$, there is some higher order cumulant which is nonzero, and hence non-Gaussian. Note also that for many specific instances of the ICA problem studied in the literature, *all* cumulants differ from those of a Gaussian [62, 107, 15].

We note one common thread that runs through all our applications of the rank 1 decomposition: the decomposition requires non-degeneracy in certain values which are essentially the tensor eigenvalues. Thus, much of the technical work in this section is aimed at establishing these spacings. The functions we're trying to space can vary – we use polynomial anti-concentration for the fully and underdetermined ICA problems, natural spacings over intervals for Gaussian mixtures, and a much harder maximum spacing calculation underpins the fast recursive fully determined algorithm. In all cases, we have to use randomness to provide the needed anti-concentration.

We remark that apart from direct practical interest of ICA in signal recovery, recently some new applications of ICA as an algorithmic primitive have been discovered. Anderson et al. [12] reduce some special cases of the problem of learning a convex body (coming from a class of convex bodies such as simplices), given uniformly distributed samples from the body, to fully-determined ICA. Anderson et al. [13] solve the problem of learning Gaussian mixture models in regimes for which there were previously no efficient algorithms known. This is done by reduction to underdetermined ICA using the results of our paper.

5.2 Subspace Juntas

5.2.1 Overview

To state our results formally, we need to define the distance of a distribution from a Gaussian via moments. Let Γ^n be the standard Gaussian distribution over $\mathbb{R}^n$ and γ_m denote the

m^{th} moment of a standard Gaussian random variable: $\gamma_m = (m-1)!!$ when m is even and 0 when m is odd.

The *m^{th}-moment distance* of two distributions F, G over $\mathbb{R}^n$ is defined as

$$d_m(F,G) = \max_{\|u\|=1} \left|\mathbb{E}_F\left((x^{\mathrm{T}}u)^m\right) - \mathbb{E}_G\left((x^{\mathrm{T}}u)^m\right)\right| = \|M_F^m - M_G^m\|_2 .$$

We say that a distribution F over $\mathbb{R}^k$ is (m,η)-*moment-distinguishable* along unit vector $u \in \mathbb{R}^k$, if either there exists $j \leq m$:

$$\left|\mathbb{E}_F\left((x^T u)^j\right) - \gamma_j\right| \geq \eta$$

or there exist unit vectors $\{v_1, \ldots, v_t\} \subset u^\perp$ where $t \leq m$ such that

$$\left|\mathbb{E}_F\left((x^T u)^{m-t}\Pi_{i=1}^t (x^T v_i)\right) - \mathbb{E}_F\left((x^T u)^{m-t}\right)\mathbb{E}\left(\Pi_{i=1}^t (x^T v_i)\right)\right| \geq \eta.$$

In words, F differs from a Gaussian either along some direction u, or by exhibiting a correlation between its marginal along u and vectors orthogonal to u (for a Gaussian such subsets have zero correlation). The rationale for this definition is that if two continuous distributions are identical (or close) in many moments, then one would expect them to be close in L^1 distance. For example, the following holds for one-dimensional logconcave distributions via an explicit bound on the number of moments required.

Lemma 5.2.1 (L^1 distance from Gaussian)**.** *Fix m and $\epsilon > 0$. Let $f : \mathbb{R} \to \mathbb{R}$ be an isotropic logconcave density, whose first m moments satisfy $|\mathbb{E}_f(x^m) - \gamma^m| < \epsilon$, then:*

$$\|f - g\|_1 \leq \left(\frac{c}{m^{1/8}} + c' m e^m \epsilon^2\right)^{1/2} \log m \leq c\left(\frac{\log m}{m^{1/16}} + \epsilon m e^m\right)$$

We are now ready to state our first main result: we can efficiently factorize distributions assuming the distribution on the relevant subspace is moment-distinguishable and the distribution on the irrelevant noisy attributes is some Gaussian. In what follows, it might be illustrative to regard k as a constant independent of n. Let $C_F(n,m,\epsilon)$ be the number of samples needed to estimate each entry of the m^{th} moment tensor of F to within additive error ϵ and M be an upper bound on the m^{th} moment along any direction.

Theorem 5.2.2 (Factoring, Gaussian noise)**.** *Let $F = F_V F_W$ be a distribution over $\mathbb{R}^n$ where V is a subspace of dimension k, and $F_W = \Gamma^{n-k}$. Suppose that F_V is (m, η)-moment-distinguishable for each unit vector $u \in V$. Then for any $\epsilon, \delta \geq 0$, in time $C_F(n, m, \epsilon)\text{poly}(n, \eta, 1/\epsilon, \log(1/\delta), M)$, Algorithm **FactorUnderGaussian** finds a subspace U of dimension at most k such that for $j \leq m$, $d_j(F, F_U F_{U^\perp}) \leq j(M + \gamma_j)\epsilon$ with probability at least $1 - \delta$. In addition, for any vector in $u \in U$, $\|\pi_V(u)\| \geq 1 - \epsilon$.*

Next we turn to learning. For a distribution F and a k-dimensional concept class $\mathcal{H}$, we say that the triple $(k, F, \mathcal{H})$ is (m, η)-*moment-learnable* if:

1. $F = F_V F_W$ is a factorizable distribution with $dim(V) = k$.
2. $\mathcal{H}$ is a set of k-subspace juntas whose relevant subspaces are contained in V.
3. For $\ell \in \mathcal{H}$ with minimal (with respect to dimension) relevant subspace $P \subseteq V$, for each unit vector $u \in P$, either F_V or F_V^+ (the distribution over the positive samples) is (m, η)-moment distinguishable along u.

In words, the third condition says that if F_V resembles a Gaussian in its first m moments along every direction, then F_V^+ does not. We will see examples of concept classes and distributions for which m is bounded under this definition. Indeed, we conjecture that a concept class $\mathcal{H}$ with bounded VC dimension d is (m, η) moment-learnable where m depends only on d and η.

To state our learning guarantee, we need one more definition: A triple $(k, F, \mathcal{H})$ is called *robust* if for any subspace U of dimension at most k with orthonormal basis $\{u_i\}$ where $\left|u_i^T \pi_V(u_i)\right| \geq 1 - \epsilon$, then $\ell(\pi_U(x))$ labels correctly $1 - g(\epsilon)$ fraction of $\mathbb{R}^n$ under F where $g(\epsilon) < \epsilon^c$ for constant $c > 0$ and sufficiently small ϵ. The definition requires the distribution F and labeling function ℓ to be robust under small perturbations of the relevant subspace. Once we identify the relevant subspace approximately, we can project samples to it and use an algorithm that can learn ℓ in spite of a $g(\epsilon)$ fraction of noisy labels.

Theorem 5.2.3 (Learning, Gaussian noise)**.** *Let $\epsilon, \delta > 0$, let $\ell \in \mathcal{H}$ where $(k, F, \mathcal{H})$ is (m, η)-moment-learnable and robust, and let $F_W = \Gamma^{n-k}$ be Gaussian. Suppose that we*

are given labeled examples from F, then Algorithm ***LearnUnderGaussian*** *identifies a subspace U and a hypothesis h such that h correctly classifies $1-\epsilon$ of F according to ℓ with probability at least $1-\delta$. The time and sample complexity of the algorithm are bounded by $T(k,\epsilon)+C_F(n,m,\epsilon)\mathrm{poly}(n,\eta,k,1/\epsilon,\log(1/\delta),M)$ where T is the complexity of learning the k-dimensional concept class $\mathcal{H}$.*

We note here that for a concept class of VC-dimension d, a standard reduction implies that the complexity of learning with ϵ arbitrary noise is at most $(2/\epsilon)^{O(d\log(1/\epsilon))}$ times the complexity of learning with no noise (Proposition 5.2.15). Our algorithms run in polynomial-time in n provided $(k, F, \mathcal{H})$ satisfy the moment-learnable condition. Some special cases of this result were previously known, e.g., when F is a Gaussian and $\mathcal{H}$ is a convex concept class [85, 125]. The application of PCA to learning convex bodies in [125] can be viewed as the assertion that convex concepts in $\mathbb{R}^k$ are moment-learnable: under a Gaussian distribution, the positive distribution F^+ has variance less than 1 along any direction. The following two examples further illustrate Theorem 5.2.3.

- When the full distribution in the relevant subspace is uniform in an ellipsoid, then robust concept classes can be learned in time $T(k,\epsilon)+C_{k,\epsilon}\cdot n^2$. Here T depends on the k and concept class, and C is a constant fixed by k and ϵ and independent of the concept class. Thus we can learn general concept classes beyond convex bodies and low-degree polynomials for uniform distributions over a ball in the relevant subspace.

- When the distribution on the positive examples F^+ has bounded support, i.e., the positive labels lie in a ball of radius $r(k)$, such robust concepts can be learned in time $T(k,\epsilon)+C_{k,\epsilon}\cdot n^{O(r(k)^2)}$ for an arbitrary distribution in the relevant subspace. Previously, for logconcave F, learning an intersection of k half-spaces was known to have complexity growing as $n^{O(k)}$ [126, 84].

We will outline the development of these ideas in what follows: first, it is critical for us to establish that subspace junta type distributions follow an additive structure, and this is what Section 5.2.2 does. Then, we can apply our additive subspace decomposition, putting the pieces together with lemmas proved in the Mathematical Preliminaries chapter. We then

build up a number of areas surrounding out models – robust geometric learning, moment distance bounds and their relationship to total variation distance, and then a number of concrete applications.

5.2.2 Structure of local optima

We derive a representation formula for $f_m(u) = \mathbb{E}\left((x^T u)^m\right)$ in Lemma 5.2.5. This structural lemma essentially states that moment tensors of a factorizable distribution are in fact additive subspace tensors. Thus, applying the theory of additive subspace tensors, we know that by Lemma 4.2.1 each local optimum lies in V or W exclusively. Finding a sequence of orthogonal local optima will give us basis vectors for the relevant subspace.

For convenience we often use $u_V = \pi_V(u)$ for the projection of u onto V, u_W for the projection onto the orthogonal subspace W, and u^0 for the unit vector in the direction of u.

We may assume that $\mathbb{E}(x) = 0$: if otherwise, then we can apply a translation $x - \mathbb{E}(x)$.

Lemma 5.2.4 (Translation of product distributions)**.** *Let $x \in \mathbb{R}^n$ be a random vector drawn from $F = F_V F_W$, a product distribution. Then $x - \mathbb{E}(x)$ has a product distribution over V and W.*

Proof of Lemma 5.5.1. Take our translation $y = T_a(x) = x + a$, for Borel sets B_1 and B_2:

$$\begin{aligned}
\mathsf{Pr}(y_V \in B_1 \wedge y_W \in B_2) &= \mathsf{Pr}(x_V + a_V \in B_1 \wedge x_W + a_W \in B_2) \\
&= \mathsf{Pr}(x_V \in B_1 - a_V \wedge x_W \in B_2 - a_W) \\
&= \mathsf{Pr}(x_V \in B_1 - a_V)\,\mathsf{Pr}(x_W \in B_2 - a_W) \\
&= \mathsf{Pr}(y_V \in B_1)\,\mathsf{Pr}(y_W \in B_2)
\end{aligned}$$

□

We can combine this with a linear transformation to obtain an isotropic distribution, given by $y = \Sigma^{-1/2}(x - \mu)$ where μ is the expectation vector. This simplifies subsequent calculations because the covariance matrix for y is I_n. The following lemma, inspired by [62, 55, 117], provides the main insight for the structural theorem.

Lemma 5.2.5 (Representation of f_m)**.** *Let $F = F_V F_W$. Suppose that x has the same j^{th} moments as a Gaussian for all integers $j < m$, then for $u \in \mathbb{S}^{n-1}$:*

$$f_m(u) = \|u_V\|^m \left(\mathbb{E}\left((x_V^{\mathrm{T}} u_V^0)^m\right) - \gamma_m\right) + \|u_W\|^m \left(\mathbb{E}\left((x_W^{\mathrm{T}} u_W^0)^m\right) - \gamma_m\right) + \gamma_m$$

Proof of Lemma 5.2.5. Consider the case when m is odd:

$$\begin{aligned}
f_m(u) &= \mathbb{E}\left((x^T u)^m\right) \\
&= \mathbb{E}\left(\left((x_V + x_W)^{\mathrm{T}}(u_V + u_W)\right)^m\right) \\
&= \mathbb{E}\left((x_V^T u_V)^m\right) + \mathbb{E}\left((x_W^T u_W)^m\right) + \sum_{i=1}^{m-1} \binom{m}{i} \mathbb{E}\left((x_V^T u_V)^i (x_W^T u_W)^{m-i}\right) \\
&= \mathbb{E}\left((x_V^T u_V)^m\right) + \mathbb{E}\left((x_W^T u_W)^m\right) + \sum_{i=1}^{m-1} \binom{m}{i} \mathbb{E}\left((x_V^T u_V)^i\right) \mathbb{E}\left((x_W^T u_W)^{m-i}\right)
\end{aligned}$$

The last line follows by applying the independence of random variables which depend only on the V and W subspaces. Each term in the last sum contains an odd moment of a Gaussian, hence:

$$f_m(u) = \|u_V\|^m \mathbb{E}\left((x_V^{\mathrm{T}} u_V^0)^m\right) + \|u_W\|^m \mathbb{E}\left((x_W^{\mathrm{T}} u_W^0)^m\right).$$

When m is even, we need the following formula:

$$\sum_{i=0}^{m} \binom{m}{i} \|u_V\|^i \|u_W\|^{m-i} \gamma_i \gamma_{m-i} = \gamma_m$$

This follows from $\mathbb{E}\left((aX + bY)^m\right) = \gamma_m$ where $a^2 + b^2 = 1$ and X and Y are independent standard normal variables:

$$\begin{aligned}
f_m(u) &= \sum_{i=0}^{m} \binom{m}{i} \|u_V\|^i \|u_W\|^{m-i} \mathbb{E}\left((x_V^T u_V^0)^i\right) \mathbb{E}\left((x_W^T u_W^0)^{m-i}\right) \\
&= \|u_V\|^m \mathbb{E}\left((x^{\mathrm{T}} u_V^0)^m\right) + \|u_W\|^m \mathbb{E}\left((x^{\mathrm{T}} u_V^0)^m\right) + \sum_{i=1}^{m-1} \binom{m}{i} \|u_V\|^i \|u_W\|^{m-i} \gamma_i \gamma_{m-i} \\
&= \|u_V\|^m \left(\mathbb{E}\left((x^{\mathrm{T}} u_V^0)^m\right) - \gamma_m\right) + \|u_W\|^m \left(\mathbb{E}\left((x^{\mathrm{T}} u_V^0)^m\right) - \gamma_m\right) + \sum_{i=0}^{m} \binom{m}{i} \|u_V\|^i \|u_W\|^{m-i} \gamma_i \gamma_{m-i} \\
&= \|u_V\|^m \left(\mathbb{E}\left((x^{\mathrm{T}} u_V^0)^m\right) - \gamma_m\right) + \|u_W\|^m \left(\mathbb{E}\left((x^{\mathrm{T}} u_V^0)^m\right) - \gamma_m\right) + \gamma_m
\end{aligned}$$

□

5.2.3 Algorithms

The previous structural lemma allows us to apply the additive subspace decomposition, but there remain a number of algorithmic considerations particular to this specific problem that we must resolve. Unfortunately, there is one more complication in using moment tensors – algorithm **FindBasis** does not suffice on its own. Although every direction orthogonal to $v \in B$ vanishes, it is possible that there are some accidental directions u which are correlated with B. The next procedure identifies such directions.

Algorithm 3 ExtendBasis

Input: Moment bound m, distribution F, candidate vectors S and non-Gaussian directions T.

$S' \leftarrow S$, $j \leftarrow 2$.
while $|S'| < k$ and $j \leq m$ **do**
  for each choice (with repetition) $\{v_1, \ldots, v_l\} \subseteq S'$ where $1 \leq l < j$. **do**
    Compute the $(j-l)$ tensor $M_{l,j}^{S',T}$ so that for any $u \in (S' \cap T)^{\perp}$,

$$g(u) = \mathbb{E}\left((x^{\mathrm{T}}u)^{j-l}\prod(x^{\mathrm{T}}v_t)\right) - \mathbb{E}\left((x^{\mathrm{T}}u)^{j-l}\right)\mathbb{E}\left(\prod(x^{\mathrm{T}}v_t)\right) = M_{l,j}^{S',T}(u,..,u,v_1,\ldots,v_l).$$

    if $g(u) \equiv 0$ **then**
      Continue with next choice of $\{v_i\}$.
    else
      if $g(u) > 0$ for any u **then**
        Compute a local maximum u^* to g starting with $u/\|u\|$.
      else
        Compute a local minimum u^* to g starting with $u/\|u\|$.
      $S' \leftarrow S' \cup \{u^*\}$ and restart the while loop with $j = 3$.
  $j \leftarrow j + 1$.
return S'.

Theorem 5.2.6 (Basis Extension)**.** *The output S' of **ExtendBasis** on input $S \subseteq V, T \subseteq W$ satisfies:*

1. *$S \subseteq S' \subseteq V$.*

2. *For $\{v_t\} \subset S'$ and $\{u_i\} \subset (S')^{\perp}$:*

$$\mathbb{E}\left(\prod_{i=1}^{j-l}(x^{\mathrm{T}}u_i)\prod_{t=1}^{l}(x^{\mathrm{T}}v_t)\right) = \mathbb{E}\left(\prod_{i=1}^{j-l}(x^{\mathrm{T}}u_i)\right)\mathbb{E}\left(\prod_{t=1}^{l}(x^{\mathrm{T}}v_t)\right)$$

The next theorem states that **ExtendBasis** finds all vectors which are correlated with

$S \subseteq V$, and that all remaining vectors at the end of the algorithm are uncorrelated up to the m^{th} moment.

Proof of Theorem 5.2.6. The Schwartz-Zippel lemma returns a correct decision at every iteration (there are at most n^k of these, so if we pick our domain to be of size $2n^k$ and run $O(\log n^k/\delta)$ iterations each time, then we have a correct decision for all iterations with probability at least $1-\delta$.

Let u^* be a local maximum found by **ExtendBasis** using the j^{th} moment. Consider the $\{v_1, \ldots, v_l\}$ where u^* was found.

$$\begin{aligned} g(u) &= \mathbb{E}\left((x^{\mathrm{T}}u)^{(j-l)}\prod_{t=1}^{l}(x^{\mathrm{T}}v_t)\right) - \mathbb{E}\left((x^{\mathrm{T}}u)^{(j-l)}\right)\mathbb{E}\left(\prod_{t=1}^{l}(x^{\mathrm{T}}v_t)\right) \\ &= \sum_{i=0}^{j-l}\binom{j-l}{i}\mathbb{E}\left((x^{\mathrm{T}}u_W)^i\right)\left[\mathbb{E}\left((x^{\mathrm{T}}u_V)^{(j-l-i)}\prod_{t=1}^{l}(x^{\mathrm{T}}v_t)\right) - \mathbb{E}\left(\prod_{t=1}^{l}(x^{\mathrm{T}}v_t)\right)\mathbb{E}\left((x^{\mathrm{T}}u_V)^{(j-l-i)}\right)\right] \end{aligned}$$

Since u^* was found at moment j, then for all $0 < i < j-l$:

$$\mathbb{E}\left((x^{\mathrm{T}}u_V)^{(j-l-i)}\prod_{t=1}^{l}(x^{\mathrm{T}}v_t)\right) = \mathbb{E}\left(\prod_{t=1}^{l}(x^{\mathrm{T}}v_t)\right)\mathbb{E}\left((x^{\mathrm{T}}u_V)^{(j-l-i)}\right)$$

Only the first and last terms survive:

$$\begin{aligned} g(u) &= \mathbb{E}\left((x^{\mathrm{T}}u_V)^{(j-l)}\prod_{t=1}^{l}(x^{\mathrm{T}}v_t)\right) - \mathbb{E}\left(\prod_{t=1}^{l}(x^{\mathrm{T}}v_t)\right)\mathbb{E}\left((x^{\mathrm{T}}u_V)^{(j-l)}\right) \\ &= \|u_V\|^{j-l}\left[\mathbb{E}\left((x^{\mathrm{T}}u_V^0)^{(j-l)}\prod_{t=1}^{l}(x^{\mathrm{T}}v_t)\right) - \mathbb{E}\left(\prod_{t=1}^{l}(x^{\mathrm{T}}v_t)\right)\mathbb{E}\left((x^{\mathrm{T}}u_V^0)^{(j-l)}\right)\right] \end{aligned}$$

Having a positive local maximum implies that $\|u_V\| = 1$.

For the second part of this lemma: we already know that all the remaining vectors must have Gaussian moments. Fix $j \leq m$ and a choice of $\{v_1, \ldots v_l\}$ from S' and consider the symmetric tensor $\hat{T}$ represented by $f(u) - \mathbb{E}\left((x^{\mathrm{T}}u)^{j-l}\right)\mathbb{E}\left(\prod_{t=1}^{l}(x^{\mathrm{T}}v_t)\right)$. We require the following claim for symmetric tensors where for any permutation $\sigma : [m] \to [m]$:

$$\mathbb{E}\left(\Pi_{k=1}^{m}x_{i_k}\right) = \mathbb{E}\left(\Pi_{k=1}^{m}x_{i_{\sigma(k)}}\right).$$

Using Claim 3.2.10:

$$\max_{\|u\|=1}\hat{T}(u,\ldots,u) = \max_{\|u_1\|=1,\ldots,\|u_{j-l}\|=1}\hat{T}(u_1,\ldots,u_{j-l})$$

In particular, there exists $\{u_i\}$ such that

$$\mathbb{E}\left(\prod_{i=1}^{j-l}(x^{\mathrm{T}}u_i)\prod_{t=1}^{l}(x^{\mathrm{T}}v_t)\right) > \mathbb{E}\left(\prod_{i=1}^{j-l}(x^{\mathrm{T}}u_i)\right)\mathbb{E}\left(\prod_{t=1}^{l}(x^{\mathrm{T}}v_t)\right)$$

if and only if there exists u such that

$$\mathbb{E}\left((x^{\mathrm{T}}u)^{j-l}\prod_{t=1}^{l}(x^{\mathrm{T}}v_t)\right) > \mathbb{E}\left((x^{\mathrm{T}}u)^{j-l}\right)\mathbb{E}\left(\prod_{t=1}^{l}(x^{\mathrm{T}}v_t)\right)$$

But at the end of the algorithm, we know that there are no such u, hence there can be no such u_i either. Thus, we can factor any $u \notin S'$ through the expectations which contain only v_i from S'. □

We now give a complete algorithm assuming F_W is a Gaussian, assuming we only have access to F through samples (not exact moment tensors). The main difficulty is handling the error accumulation over multiple iterations, as in each round we can only hope to approximately compute moments and find approximate local optima. The idea is that **FindBasis** and **ExtendBasis** find vectors where $\mathbb{E}\left((x^T u)^m\right) \neq \gamma_m$. If F_W is Gaussian, our algorithms only find directions in V. Thus, the error accumulates over only k steps, and the total error depends on k rather than n.

Using **LocalOpt**, we have an algorithm for factoring (Problem 5). To deal with the errors introduced by sampling and approximate local optima, we replace the Schwartz-Zippel step in **FindBasis** with the robust version in Lemma 3.1.4, where we set the error parameter of the robust Schwartz-Zippel lemma to be $(\eta - \|M^m\|_2 \epsilon)/n^m$.

Algorithm 4 FactorUnderGaussian

Input: Highest moment m, distribution F.

1: $B \leftarrow$ ***FindBasis*$(\boldsymbol{m}, \boldsymbol{F})$**.
2: $U \leftarrow$ ***ExtendBasis*$(\boldsymbol{m}, \boldsymbol{F}, \boldsymbol{B}, \boldsymbol{\phi})$**.
3: **return** U

Proof of Theorem 5.2.2. We choose ϵ_1 (the first step local iteration) to be:

$$(\epsilon_1)^{\left(\frac{1}{16}\right)^k} \leq \min\{\epsilon, \eta - \|M^m\|_2 \epsilon\}$$

where $\|M^m\|_2$ is the 2-norm (spectral norm) of the m^{th} moment tensor. We take enough samples so that each estimated moment in W is within $\min(\epsilon_1, \eta - \|M^m\|_2)/n^m)$ of the

Gaussian moment, and every moment in V is off by at most $\min(\epsilon_1/2, \eta/2)$. In particular, note that all sampled gradients and Hessian matrices take a value which differ by no more than $\epsilon_1/2$ from their true values. Thus, we can simply absorb this as part of the error arising from local search. Also, this gives us an upper bound on sample complexity – the number of samples it takes to estimate the m^{th} moments of a Gaussian distribution to accuracy ϵ in $\mathbb{R}^n$ is given as $C_m \epsilon^{-2} n^{m/2} \log n$ [67], which when evaluated becomes $n^{O(m)}$.

At each iteration of the algorithm, we run the Robust Schwartz-Zippel test $\log(k/\delta)$ times with Schwartz-Zippel parameter $\eta - \|M^m\|_2 \epsilon$. With probability at least $1-\delta$, either each iteration produces a u, where $\left|\mathbb{E}\left((x^T u)^m\right) - \gamma_m\right| \geq \eta - \|M^m\|_2 \epsilon$ or we correctly deduce that there are no more directions whose moments differ from a Gaussian by more than $(\eta - \|M^m\|_2 \epsilon)/n^m$. In the latter case, by moment distinguishability, every vector in P, the minimally relevant subspace, has large projection on the basis $\{u_i\}$.

In the former case, we know that every unit vector in $\{u_i\}^\perp$ with projection at least $1-\epsilon$ takes value which is bounded away from γ_m by at least $\eta - \|M^m\|_2 \epsilon$, thus every such vector is still moment distinguishable. Applying Theorem 4.2.8 then, we sequentially generate a sequence of at most k orthogonal u_i such that:

$$|\langle u_i, \pi_V(u_i)\rangle| \geq 1 - (\epsilon_1)^{(1/16)^i}$$

We need to show that in addition $d_m(F, \hat{F}_U \hat{F}_{U^\perp}) \leq \epsilon$. Let $F' = \hat{F}_U \hat{F}_{U^\perp}$: the moment-distance between the true and sampled distributions differ by at most ϵ_1, it suffices for us to prove that $d_m(F, F') \leq \epsilon$. To this end, we will apply the representation formula to F' for some fixed unit vector u. As before, we have:

$$\begin{aligned}
\mathbb{E}_{F'}\left((x^T u)^m\right) &= \mathbb{E}_{F'}\left((x^T u_U)^m\right)\right) + \mathbb{E}_{F'}\left((x^T u_{U^\perp})^m\right) - \gamma_m \|u_U\|^m - \gamma_m \|u_{U^\perp}\|^m + \gamma_m \\
&= \mathbb{E}_{F}\left((x^T u_U)^m\right)\right) + \mathbb{E}_{F}\left((x^T u_{U^\perp})^m\right) - \gamma_m \|u_U\|^m - \gamma_m \|u_{U^\perp}\|^m + \gamma_m
\end{aligned}$$

Hence, comparing with a similar expression for $\mathbb{E}_F\left((x^T u)^m\right)$:

$$\begin{aligned}
\left|\mathbb{E}_{F'}\left((x^T u)^m\right) - \mathbb{E}_F\left((x^T u)^m\right)\right| \leq &\left|\mathbb{E}_F\left((x^T u_U)^m\right)\right) - \mathbb{E}_F\left((x^T u_V)^m\right)\right| + \\
&+ \left|\mathbb{E}_{F'}\left((x^T u_{U^\perp})^m\right) - \mathbb{E}_F\left((x^T u_{V^\perp})^m\right)\right| \\
&+ \left|\gamma_m \|u_U\|^m - \|u_V\|^m\right| + \gamma_m \left|\|u_U\|^m - \|u_{U^\perp}\|\right|
\end{aligned}$$

Now, viewing $\mathbb{E}_F\left((x^T u)^m\right)$ as the tensor applied to u, we see that we can bound these terms by the tensor spectral norm:

$$\left|\mathbb{E}_{F'}\left((x^T u)^m\right) - \mathbb{E}_F\left((x^T u)^m\right)\right| \leq \left(\|M^m\|_2 + m\gamma_m\right)\|u_U - u_V\| + \left(\|M^m\|_2 + m\gamma_m\right)\|u_{U^\perp} - u_{V^\perp}\|$$

By choice of U, we have $\|u_U - u_V\| \leq \epsilon$, and similarly for the othogonal component, thus we have our bound. □

We now apply these methods to learning the concept class $\mathcal{H}$ (Problem 6). After applying an isotropic transformation, F will have Gaussian moments in every direction orthogonal to V, and hence the output basis of **FindBasis** and **ExtendBasis** returns only vectors in the V subspace. The proof of this algorithm is straightforward given the proof of the

Algorithm 5 LearnUnderGaussian

Input: Highest moment m, distribution F.

1: $B_1 \leftarrow \boldsymbol{FindBasis(m, F)}$.
2: $B_2 \leftarrow \boldsymbol{FindBasis(m, F^+)}$ on the space orthogonal to B_1.
3: Alternately run ***ExtendBasis*** on F and F^+ to find a basis U of size at most k. Extend this to a basis of dimension k.
4: Draw sufficient samples S to learn $\mathcal{H}$ on this k dimensional subspace. Project S to span (U).
5: Learn $\mathcal{H}$ over U.

factoring algorithm under Gaussian noise and our robustness assumptions. We will use the following proposition on robust learnability (see e.g., [16]).

Proposition 5.2.7 (VC dimension)**.** *Let $\mathcal{H}$ be a hypothesis class with VC dimension d. Let $\ell \in \mathcal{H}$ be a subspace junta with relevant subspace V, where $dim(V) = k$. Let U be a k dimensional subspace where $\ell(\pi_U)$ labels a $1-\epsilon$ fraction of points correctly. Then we can learn ℓ with sample complexity $(1/\epsilon)^{c_2 d \log(1/\epsilon) + c_2 \log(2/\delta)}$ with probability at $1-\delta$.*

Proof of Theorem 5.2.3. $\mathcal{H}$ is *robust*; by assumption there exists ϵ' which is polynomial in ϵ such that $\epsilon' + g(\epsilon') \leq \epsilon/2$. We will take this ϵ' and will use the following ϵ_1 for all our calls to **LocalOpt**:

$$(\epsilon_1)^{\left(\frac{1}{16}\right)^k} \leq \min\{\epsilon', \eta - \|M^m\|_2\,\epsilon\}$$

Under these parameters, the proof for the factoring steps of Lines 1-3 are as in **FactorUnderGaussian**. Thus with probability at least $1 - \delta$ we will obtain an orthonormal basis $\{u_i\}$ where $|\langle u_i, \pi_V(u_i)\rangle| \geq 1 - (\epsilon_1)^{(1/16)^i}$.

By moment learnability, the set of $\{u_i\}$ discovered is approximately a basis for P, the minimal dimension relevant subspace. By our choice of ϵ_1 above, we satisfy the robustness condition, i.e., $\epsilon_1^{16^k} \leq \epsilon'$, in which case only $\epsilon/2$ fraction of the points are mislabeled over span $(\{u_i\})$. Finally, we allow the remaining $\epsilon/2$ error to the learning algorithm, to obtain an output hypothesis which correctly labels $1 - \epsilon$ fraction of F. □

5.2.4 Moments

In this section, we highlight some further consequences and subtleties of using moments in algorithms. The use of moments is a very natural way of studying random variables. For example, the inequalities of Markov, Chebyshev and Chernoff are statements about the relationship between a finite sequence of moments and the tail of a distribution. If we consider an infinite sequence of moments, often these will determine the distribution uniquely (the moments problem).

One of the critical terms in the runtime given in our main theorems is $C_F(m, \epsilon)$: the sample complexity of approximating the m^{th} moment tensor of distribution F to within accuracy ϵ (in the moment metric above). The competitiveness of our algorithm with other learning algorithms depends on the number of moments we need, and the number of samples we need to attain the required accuracy. This latter problem is well-studied, and there is an impressive body of literature surrounding it. In particular, when $m = 2$, the problem is of interest to random matrices community, who have provided strong bounds in a number of important cases. We will provide a brief overview of these results for logconcave distributions, but this by no means is intended to be a comprehensive survey of the literature! When the distribution F is isotropic and almost surely supported in a ball of radius $O(\sqrt{n})$, Rudelson [114] gave a very strong bound on $C_F(n, \epsilon)$ to achieve the

following guarantee for convex bodies:

$$\mathbb{E}\left(\left\|\frac{1}{N}\sum_{i=1}^{N} x_i x_i^T - I\right\|\right) \leq \epsilon.$$

Rudelson required only $O(n\log(n))$ samples when F is almost surely supported on a ball of radius $O(\sqrt{n})$, and where the constant is dependent on ϵ. Adamczak et al. [2] were able to improve this bound of $O(n)$ samples. Their assumptions were support on a ball of radius $O(\sqrt{n})$ as before, and a subexponential moment condition:

$$\sup_{\|v\|=1} \mathbb{E}\left((x^T v)^p\right)^{1/p} = O(p)$$

As an application, they showed that logconcave distributions satisfy these assumptions, and thus their covariance matrices can be sampled very efficiently. Subsequent work by Vershynin and collaborators [121, 131] has broadened the class of efficiently sampleable covariance matrices to distributions where $2+\epsilon$ moments exist and also to distributions where the m^{th} moment is bounded by K^m for some constant K.

For higher moments, there is the result of Guedon and Rudelson [67], which gives the sample complexity of sampling for higher moments of logconcave distributions. Their result is that $O(n^{m/2}\log(n))$ samples are necessary to approximate moments in all directions up to an $1+\epsilon$ factor. In particular, this leads to the observation that explicitly computing a sample moment tensor from $n^{m/2}$ samples is actually less efficient than simply storing the points, computing the inner products to the appropriate powers and summing. This last result is used in our applications in Section 5.2.5, as it allows us to handle many distributions efficiently, including Gaussians and uniform distributions over convex bodies.

In our algorithms, we terminate if all remaining directions are Gaussian in the m^{th} moment (for some fixed m). We would like a guarantee that when we do this, that the random variable is in fact very close to being Gaussian. What follows is a set of results which quantify this idea. We first restrict ourselves to one random variable to introduce the analytic tools we need. In what follows, we use the following normalisation for our fourier transforms in $\mathbb{R}^n$:

$$\hat{f}(\xi) = \int e^{i\xi\cdot x} f(x) dx$$

This implies that the Parseval/Plancherel theorem takes the following form:

$$\int |f(x)|^2 \, dx = \frac{1}{(2\pi)^n} \int \left| \hat{f}(\xi) \right|^2 d\xi$$

for $f \in L^2(\mathbb{R}^n)$.

The core of the proof is the following statement, whose proof employs Fourier analytic techniques. We need the following standard theorem on characteristic functions (see for example [118]):

Theorem 5.2.8 (Characteristic functions)**.** *Let ξ be a random variable with distribution function $F = F(x)$ and $\phi(t) = \mathbb{E}\left(e^{it\xi}\right)$ its characteristic function. Let $\mathbb{E}\left(|\xi|^n\right) < \infty$ for some $n \geq 1$, then $\phi^{(r)}$ exists for all $r \leq n$ and*

$$\phi^{(r)}(t) = \int (ix)^r e^{itx} dF(x)$$

Moreover, we have an expression for the derivatives at 0:

$$\mathbb{E}\left(\xi^r\right) = \frac{\phi^{(r)}(0)}{i^r}$$

And finally we have the following Taylor series estimate with error:

$$\phi(t) = \sum_{r=0}^{n} \frac{(it)^r}{r!} \mathbb{E}\left(\xi^r\right) + \frac{(it)^n}{n!} \epsilon_n(t)$$

where the error term $\epsilon_n(t) \to 0$ as $n \to \infty$ and is bounded:

$$|\epsilon_n(t)| \leq 3\mathbb{E}\left(|\xi|^n\right)$$

Now:

Lemma 5.2.9 (L^2 distance from a Gaussian)**.** *Let $f \in L^2(\mathbb{R})$ be a probability density over $\mathbb{R}$ whose first m moments match those of a standard Gaussian (whose probability density we will denote g). Suppose that the Fourier transform $\hat{f}$ satisifies a tail bound that $\left|\hat{f}(\xi)\right| < c/|\xi|$ for some $c > 0$, then:*

$$\int_{\mathbb{R}} |f(x) - g(x)|^2 \, dx \leq \frac{c'}{m^{1/8}}$$

Proof. We will assume for the sake of simplicity that m is even. By Parseval's formula, we have:

$$\int |f(x) - g(x)|^2 \, dx = \frac{1}{\sqrt{2\pi}} \int |(f-g)\hat{}(\xi)|^2 \, d\xi$$

Both f and g have tail bounds: f by hypothesis, and g because the Fourier transform of a Gaussian is still a Gaussian. Thus if we truncate the tails in an interval $[-L, L]$ where $L = m^{1/8}$:

$$\begin{aligned}\int_{\mathbb{R}/[-L,L]} \left|\hat{f}(\xi)\right|^2 d\xi &\leq 2\int_L^\infty \frac{1}{\xi^2} d\xi \\ &\leq \frac{4}{L}\end{aligned}$$

The Fourier transform of a Gaussian is a Gaussian, and by applying a standard Gaussian tail bound [61]:

$$\frac{1}{\sqrt{2\pi}} \int_x^\infty e^{-t^2/2} dt \leq \left(\frac{1}{x}\right) \frac{e^{-x^2/2}}{\sqrt{2\pi}}$$

We can then combine these estimates using the triangle inequality:

$$\begin{aligned}\int_{\mathbb{R}/[-L,L]} \left|f \hat{-} g(\xi)\right|^2 d\xi &\leq \int_{\mathbb{R}/[-L,L]} \left|\hat{f}(\xi)\right|^2 + |\hat{g}(\xi)|^2 \, d\xi \\ &\leq \frac{6}{L}\end{aligned}$$

In the interval $[-L, L]$, we now apply Theorem 5.2.8:

$$\begin{aligned}(\hat{f} - \hat{g})(\xi) &= \sum_{k=0}^m \frac{\mathbb{E}_f\left(x^k\right) - \mathbb{E}_g\left(x^k\right)}{k!} (i\xi)^k + (\epsilon_f(t) - \epsilon_g(t)) \frac{(i\xi)^m}{m!} \\ &= (\epsilon_f(t) - \epsilon_g(t)) \frac{(i\xi)^m}{m!}\end{aligned}$$

Now we can bound the integral:

$$\begin{aligned}\int_{-L}^L |(f-g)\hat{}(\xi)|^2 \, d\xi &\leq \int_{-L}^L \left|(\epsilon_f(\xi) - \epsilon_g(\xi)) \frac{(i\xi)^m}{m!}\right|^2 d\xi \\ &\leq 6\left(\frac{\mathbb{E}\left(x^m\right)}{m!}\right)^2 \int_{-L}^L t^{2m} dt \\ &\leq \frac{6}{(2^{m/2}(m/2)!)^2} \frac{2L^{2m+1}}{2m+1} \\ &\leq \frac{12}{2m+1} \exp\left((2m+1)\log(L) - m\log(2) - m\log\left(\frac{m}{2}\right) + m\right) \\ &\leq \frac{c}{m} e^{-m}\end{aligned}$$

□

We can also give an approximate version of this theorem:

Lemma 5.2.10 (Approximate moments)**.** *Fix $\epsilon > 0$, let $f \in L^2(\mathbb{R})$ be a probability density over $\mathbb{R}$ whose first m moments satisfy:*

$$\left|\mathbb{E}_f\left(x^k\right) - \gamma_k\right| \leq \epsilon$$

Suppose that the Fourier transform $\hat{f}$ satisifies a tail bound that $\left|\hat{f}(\xi)\right| < c/|\xi|$ for some $c > 0$, then (for a standard Gaussian g);

$$\int_{\mathbb{R}} |f(x) - g(x)|^2 \, dx \leq \frac{c'}{m^{1/8}} + c'' m^2 \epsilon^2 e^m$$

Proof. We proceed as in the previous lemma. It suffices for us to bound the integral over the interval $[-L, L]$. We apply Cauchy-Schwarz for a termwise estimate.

$$\int_{-L}^{L} \left| \sum_{k=0}^{m} \frac{\mathbb{E}_f\left(x^k\right) - \mathbb{E}_g\left(x^k\right)}{k!} (i\xi)^k + (\epsilon_f(t) - \epsilon_g(t)) \frac{(i\xi)^m}{m!} \right|^2 d\xi$$
$$\leq m \int_{-L}^{L} \sum_{k=0}^{m} \left(\frac{\mathbb{E}_f\left(x^k\right) - \mathbb{E}_g\left(x^k\right)}{k!} \xi^k \right)^2 + \left((\epsilon_f(t) - \epsilon_g(t)) \frac{\xi^m}{m!} \right)^2 d\xi$$

We can now partition the moments into powers of 2, so consider the moments where $k \in [m/2^{i+2}, m/2^i]$: the integral of each contributing term is now:

$$\int_{-L}^{L} \left(\frac{\mathbb{E}_f\left(x^k\right) - \mathbb{E}_g\left(x^k\right)}{k!} \xi^k \right)^2 d\xi = \frac{2(\mathbb{E}_f\left(x^k\right) - \mathbb{E}_g\left(x^k\right))^2 L^{2k+1}}{(2k+1)k!}$$
$$\leq 2\left(\mathbb{E}_f\left(x^k\right) - \mathbb{E}_g\left(x^k\right)\right)^2 \exp\left(\frac{2k+1}{8} \log(m) - k \log k + k \right)$$
$$\leq 2\left(\mathbb{E}_f\left(x^k\right) - \mathbb{E}_g\left(x^k\right)\right)^2 \exp\left(\frac{(m/2^i)+2}{4} \log(m) - \frac{m}{2^{i+2}} \log\left(\frac{m}{2^{i+2}}\right) + \frac{m}{2^i} \right)$$
$$\leq 2\left(\mathbb{E}_f\left(x^k\right) - \mathbb{E}_g\left(x^k\right)\right)^2 \exp\left(\frac{m}{2^{i-1}}\right)$$

□

Both of our lemmas so far in this section use a tail bound for the Fourier transform. One way to obtain such a tail-bound is to examine logconcave probability densities:

Lemma 5.2.11 (Log-concave densities)**.** *Let $f : \mathbb{R} \to \mathbb{R}$ be a logconcave density which is isotropic and differentiable, then $\left|\hat{f}(\xi)\right| \leq 2/|\xi|$.*

Proof. We start by bounding the magnitude of the Fourier transform by the integral of the derivative.

$$\begin{aligned}\hat{f}(\xi) &= \int_{\mathbb{R}} e^{i\xi x} f(x) dx \\ &= \int_{\mathbb{R}} \frac{1}{i\xi} \frac{d}{dx} e^{i\xi x} f(x) dx \\ &= \int_{\mathbb{R}} \frac{1}{i\xi} e^{i\xi x} \frac{df(x)}{dx} dx\end{aligned}$$

where the third line follows by integration by parts and noting that in the limit $f(x) \to 0$ as $x \to \pm\infty$. This allows us to bound $\hat{f}(\xi)$:

$$\left|\hat{f}(\xi)\right| \leq \frac{1}{|\xi|} \int_{\mathbb{R}} \left|f'(x)\right| dx$$

Let us now turn to logconcave densities. Since f is logconcave, we can write it as $f(x) = e^{h(x)}$ where h is concave. Because f is a probability density, we must have $h(x) \to -\infty$ as $x \to \pm\infty$, in which case since h is concave there exists a unique interval $[a, b]$ where $h(x)$ takes a maximum. This fully determintes the sign of the derivative: $h'(x) = 0$ in this interval $h'(x) < 0$ for $x < a$ and $h'(x) > 0$ for $x > b$. The same signs pattern holds for f', as multiplication by $e^{-h(x)}$ does not change the sign. We can now compute the integral by applying the fundamental theorem of calculus:

$$\begin{aligned}\int_R \left|f'(x)\right| dx &= \int_{-\infty}^{a} f'(x)dx + \int_a^b f'(x)dx + \int_b^{\infty} -f'(x)dx \\ &= \lim_{t\to\infty} (f(a) - f(-t)) + (f(b) - f(a)) + (-f(t) + f(b)) \\ &= f(a) + f(b) \\ &= 2f(a)\end{aligned}$$

We now apply the following lemma [97], which yields the desired result.

Lemma 5.2.12 (Upper bound on logconcave functions)**.** *Let f be an isotropic logconcave density in one dimension, then $|f(x)| \leq 1$.*

□

Then as a corollary to Lemma 5.2.9:

Corollary 5.2.13 (L^2 distance for logconcave densities)**.** *Let $f : \mathbb{R} \to \mathbb{R}$ be an isotropic logconcave density whose first m moments match a Gaussian g, then:*

$$\|f - g\| \leq \frac{c}{m^{1/8}}$$

Proof. First, consider the case when $f(x)$ is differentiable. We already know that $f \in L^1(\mathbb{R})$; since $f(x)$ is bounded by 1 (Lemma 5.2.12), then we have that $f(x) \in L^2(\mathbb{R})$ because $f(x)^2 \leq |f(x)|$. We can now apply Theorem 5.2.9 with the tail bound guaranteed by Lemma 5.2.11.

For the case when $f(x)$ is *not* differentiable, we can perturb by a small Gaussian random variable: let $X \sim f$, and let $Z \sim N(0,1)$ be an independent normal variable. Fix a parameter $\tau \in [0,1]$:

$$Y_\tau = (1-\tau)X + \sqrt{2\tau + \tau^2}Z$$

is isotropic. Moreover, since this the sum of two independent logconcave random variables, its density is also logconcave. Let h_1 denote the density of $(1-\tau)X$ and h_2 the density of $\sqrt{2\tau+\tau^2}Z$, then the density of our new random variable is given by:

$$h_1 * h_2(x) = \int_{-\infty}^{\infty} h_1(x-t)h_2(t)dt$$

The convolution of these two distributions is (infinitely) differentiable because h_2 is (infinitely) differentiable:

$$\frac{d}{dx}(h_1 *) = \left(\frac{d}{dx}h_1\right) * h_2 = h_1 * \left(\frac{d}{dx}h_2\right)$$

Thus Y_τ satisfies the hypotheses of Lemma 5.2.11, and we have a tail bound for Y_τ as long as $\tau > 0$.

The first m moments of Y are also close to those of X: if we compute the j^{th} moment for example:

$$\begin{aligned}\mathbb{E}\left(Y_\tau^j\right) &= \mathbb{E}\left(\left((1-\tau)X + \sqrt{2\tau+\tau^2}Z\right)^j\right) \\ &= (1-\tau)^j\mathbb{E}\left(X^j\right) + \sum_{i=1}^{j}\binom{i}{j}(1-\tau)^j(\sqrt{2\tau+\tau^2})^{i-j}\mathbb{E}\left(X^i\right)\mathbb{E}\left(Z^{i-j}\right)\end{aligned}$$

Thus we can pick τ small enough so that:

$$\left|\mathbb{E}\left(Y_\tau^j\right) - \mathbb{E}\left(X^j\right)\right| \leq \epsilon$$

for any $\epsilon > 0$. In the proof of Lemma 5.2.9 then, instead of the moment differences from the first m terms of the characteristic function being 0, we can make them arbitrarily small by choosing smaller τ. Thus we have the conclusion of Lemma 5.2.9 for Z. To conclude, we note that:

$$\lim_{\tau \to 0} \|h_1 * h_2 - f\|_2 = 0$$

in which case, taking τ small enough allows us to apply the triangle inequality to:

$$\|f - g\| \leq \|f - h\| + \|h - g\|$$

□

We also need a lemma to convert our L^2 estimates to L^1 estimates. This is not general in possible, but since logconcave functions have exponential tailbounds:

Lemma 5.2.14 (L^2 to L^1)**.** *Let $f, g : \mathbb{R} \to \mathbb{R}$ isotropic logconcave densities such that for some $m > 0$ that:*

$$\int |f(x) - g(x)|^2 \, dx \leq \frac{1}{m}$$

then:

$$\int |f(x) - g(x)| \, dx \leq \frac{c \log(m)}{\sqrt{m}}$$

for some absolute constant $c > 0$.

Proof. Fix $L = (\frac{1}{c}) \log(m)$, then as before:

$$\int |f(x) - g(x)| \, dx = \int_{|x| \leq L} |f(x) - g(x)| \, dx + \int_{|x| > L} |f(x) - g(x)| \, dx$$

We can now use tail bound for logconcave functions over the tail [66], in particular, for isotropic logconcave random variables X in $\mathbb{R}^n$, we have (for some fixed absolute constants $c, C > 0$:

$$\mathsf{Pr}\left(\left|\|x\| - \sqrt{n}\right| \geq t\sqrt{n}\right) \leq C \exp\left(-cn^{\frac{1}{2}} \min(t, t^3)\right)$$

In one dimension, this shows that the integral of our tail is bounded by C/m (after application of triangle inequality). Now inside the interval $[-L, L]$, we will apply the Cauchy-Schwartz inequality:

$$\begin{aligned}\int_{[-L,L]} |f(x) - g(x)|\, dx &\leq \left(\int_{[-L,L]} |f(x) - g(x)|^2\, dx\right)^{1/2} \left(\int_{[-L,L]} 1 dx\right)^{1/2} \\ &\leq \frac{\sqrt{2}}{c\sqrt{m}} \log(m)\end{aligned}$$

□

Proof of Theorem 5.2.1. The proof follows from Lemma 5.2.10, Corollary 5.2.13 and Lemma 5.2.14, noting that the the technique of Corollary 5.2.13 can be applied to Lemma 5.2.10 in the same way as Lemma □

5.2.5 Examples

In this section, we give some applications of our general theorems and we some explicit bounds for moment-learnable triples and the running time of our algorithms on these triples. We make explicit in our analysis the three key contributions to runtime – how many moments are required, how efficiently these moments can be sampled, and how efficiently the hypothesis can be learned in the k-dimensional relevant subspace.

For learning over a k-dimensional subspace, we have the following proposition:

Proposition 5.2.15 (VC dimension)**.** *Let $\mathcal{H}$ be a hypothesis class with VC dimension d. Let $\ell \in \mathcal{H}$ be a subspace junta with relevant subspace V, where $dim(V) = k$. Let U be a k dimensional subspace where $\ell(\pi_U)$ labels a $1 - \epsilon$ fraction of points correctly. Then we can learn ℓ with sample complexity $(1/\epsilon)^{c_2 d \log(1/\epsilon) + c_2 \log(2/\delta)}$ with probability at $1 - \delta$.*

Proof. To come up with a hypothesis over U, we take a new set of samples S of size m and project them onto U. By robustness of $\mathcal{H}$ under F, we know that $\mathsf{Pr}\left(\ell(\pi_U(x)) = \ell(x)\right) \geq 1 - \epsilon$. Then we guess the correct labels by trying all relabelings of subsets of size ϵm. One of these relabelings will give us a labeling consistent with ℓ viewed as a function of the k-coordinates in U. For each relabeling we attempt to learn the labeling function. On the

correct relabeling, we can learn ℓ to with at most ϵ fraction of errors. By the theorem above, our total error over $\mathbb{R}^n$ is 2ϵ.

To bound m, we apply an idea from [30] via a slight extension (Theorem 5 of [16]). The required bound is $m \geq (32/\epsilon)\log(C[m]) + (32/\epsilon)\log(2/\delta)$ where $C[m]$ is the maximum number of distinct labelings obtainable using concepts in $\mathcal{H}$ over $\mathbb{R}^k$. In particular, we have $C[m] \leq \sum_{i=0}^{d} \binom{m}{i}$, whence $C[m] \leq m^d$. A computation reveals that $m \geq c(d/\epsilon)\log(1/\epsilon) + (c'/\epsilon)\log(2/\delta)$ suffices. The number of relabelings is $\binom{m}{\epsilon m}$, which is upper bounded by $(m/\epsilon)^{m\epsilon} \leq (1/\epsilon)^{c_2 d \log(1/\epsilon) + c_2 \log(2/\delta)}$. □

As mentioned previously, we can view the work of [125] as a specialization of our algorithms to the $j = m = 2$ case in **FindBasis**. We give examples here where the second moment does *not* suffice, and we must use higher moments to resolve the relevant subspace V. Our examples are: (1) hyperrectangles (cuboids) in balls, (2) subsets of balls, and (3) concepts which have compact support. In all our examples, the algorithm used is **LearnUnderGaussian**. We will prove that we can find the relevant subspaces by running **FindBasis** on either the full distribution or distribution conditioned on positive labels (the "positive" distribution).

We use the uniform distribution over a ball in $\mathbb{R}^k$ in the relevant subspace. We need the following elementary fact.

Claim 5.2.16 (Isotropic balls)**.** *Let F be the uniform distribution (with density ρ) over $\mathbb{B}_R(0) \subset \mathbb{R}^n$ where $R = \sqrt{n+2}$, then $\mathbb{E}\left((x^T u)^2\right) = 1$ for any unit vector u.*

By a hyperrectangle, we refer to a region of space which is the Cartesian product of closed intervals i.e. $S = [a_i, b_i] \times \cdots \times [a_k, b_k] \subset \mathbb{R}^k$:

Application 1 (Hyperrectangles in balls)**.** *Let $F = F_V F_W$ where F_V is a uniform distribution over a ball $\mathbb{B}$ and $k = dim(V)$, F_W is any Gaussian over $n-k$ dimensions. Let $S \subset \mathbb{B}$ denote a (hyper)rectangle in V. Take the hypothesis class $\mathcal{H} = \{(\chi_S(\pi_V))(x) : S \subset \mathbb{B}\}$ to be the set of functions which assigns positive labels to points whose projection to V lies in the interior of rectangle S.*

Proposition 5.2.17. *The triple* $(k, F, \mathcal{H})$ *as defined in Application 1 is* $(4, 6/(5k))$ *moment-learnable with time and sample complexity* $\text{poly}(k, 1/\epsilon) + C_{k,\epsilon} n^2$.

Proof. Without loss of generality, we may assume that $\mathbb{B} = \mathbb{B}_{\sqrt{n+2}}(0)$ after isotropic transformation, and that the Gaussian over F_W is a standard n-dimensional Gaussian. Furthermore, we may assume that S is centered on the origin as well (i.e. we apply Lemma 5.5.1 to the positively labeled points).

Suppose we now run **LearnUnderGaussian** on the positively labeled samples. We start with the second moment ($r = 2$) in our algorithm **FindBasis**: the second moments of a uniform distribution over a rectangle are fully determined by the second moments along the axes of the rectangle. In particular, **FindBasis** using the second moments will simply give us every axis of the rectangle where the second moment is not 1. A simple calculation of the moments of a uniform distribution over a rectangle along axis x_i where the rectangle has length $2S_i$ gives:

$$\mathbb{E}\left(x_i^2\right) = \int_{-S_i}^{S_i} x_i^2 \frac{1}{2S_i} dx_i = \frac{S_i^2}{3}.$$

Thus, using the second moment will give us all the axes of our hyperrectangle except where the rectangle has length $2S_i = 2\sqrt{3}$. Projecting orthogonally to these axes, we now consider the third moments ($r = 3$): the third moment of our uniform rectangle is clearly 0 in every direction by symmetry of the rectangle. Thus, we turn to the fourth moment – note that fixing $S_i = \sqrt{3}$ fixes the fourth moment along each axis of the rectangle, in particular:

$$\mathbb{E}\left(x_i^4\right) = \int_{-S_i}^{S_i} x_i^4 \frac{1}{2S_i} dx_i = \frac{9}{5}.$$

Unfortunately, the equality of the fourth moment along the axes of a rectangle does not necessarily imply the same fourth moment in every direction. However, iterating Lemma 5.2.5 allows us to bound the fourth moments away from the fourth moment of a Gaussian $\gamma_4 = 3$:

$$\mathbb{E}\left((x^T u)^4\right) = \left(\frac{9}{5} - \gamma_4\right) \sum_{i \in R'} u_i^4 + \gamma_4$$

where the sum is taken over directions corresponding to axes where $S_i = \sqrt{3}$. Now by applying the Lagrangian style techniques of Lemma 4.2.1, we can bound this by:

$$\mathbb{E}\left((x^T u)^4\right) \leq \gamma_4 - \frac{6}{5k}$$

Thus, we have our moment learnability using only the fourth moment! Now that we have the relevant subspace V, we can simply learn our rectangle in a dimension k space, which takes poly(k) time. Moreover, note that since all the distributions are logconcave, we can apply the moment sampling results of Guedeon and Rudelson mentioned in Section 5.2.4 – in particular, we can take the number of samples required to be $C_F(m, \epsilon) = C_\epsilon n^2$. Thus this gives a final runtime of poly$(k) + C_{k,\epsilon} n^2$ where $C_{k,\epsilon}$. □

The key point here is that we have very low polynomial dependence in n. This conforms well with our model where we think of k as being small compared to n. We can, in fact, prove a stronger result — we can always find the relevant subspace if F_V is a uniform distribution over a ball:

Application 2 (Uniform distributions over balls)**.** *Let $F = F_V F_W$ where F_V is a uniform distribution over a ball $\mathbb{B}$ and $k = dim(V)$, F_W is a Gaussian. Let $\mathcal{H}$ be a robust hypothesis class which we can learn with complexity bounded by $T(k, \epsilon)$.*

Proposition 5.2.18. *The triple $(k, F, \mathcal{H})$ as defined in Application 2 is $(4, \Omega(1))$ moment-learnable, with the time and sample complexity bounded by $T(k, \epsilon) + C_{k,\epsilon} n^2$.*

Proof. We will examine what happens when we run **FindBasis** on the full distribution (as opposed to the positive distribution in the previous example). We compute the fourth moment of a ball of radius $R = \sqrt{n+2}$. For simplicity, we will assume that $k = 2l + 1$ for

some positive integer l ie k is odd:

$$\begin{aligned}
\mathbb{E}\left(x_1^4\right) &= \int_{\mathbb{B}_R(0)} x_1^4 \rho dx \\
&= \int_{-R}^{R} \int_{\mathbb{B}^{k-1}_{\sqrt{R^2-x_1^2}}(0)} x_1^4 \rho dx_2 \cdots dx_k dx_1 \\
&= \frac{1}{\operatorname{vol}\left(\mathbb{B}_R^k(0)\right)} \int_{-R}^{R} x_1^4 \operatorname{vol}\left(\mathbb{B}^{k-1}_{\sqrt{R^2-x_1^2}}(0)\right) dx_1 \\
&= \frac{\operatorname{vol}\left(\mathbb{B}_R^{k-1}(0)\right)}{\operatorname{vol}\left(\mathbb{B}_R^k(0)\right)} \int_{-R}^{R} x_1^4 \left(1 - \frac{x_1^2}{R^2}\right)^l dx_1
\end{aligned}$$

We first examine the volume ratio: using the recurrence:

$$\operatorname{vol}\left(\mathbb{B}_R^k(0)\right) = \frac{2\pi R^2}{k} \operatorname{vol}\left(\mathbb{B}_R^{k-2}(0)\right)$$

and unrolling the recurrence, we have:

$$\begin{aligned}
\frac{\operatorname{vol}(2l)}{\operatorname{vol}(2l+1)} &= \frac{(2l+1)!!}{2R(2l)!!} \\
&= \frac{1}{2R} \frac{(2l+2)!!}{(l+1)!2^{l+1} l! 2^l} \\
&= \frac{1}{2R} \frac{(2l+2)!!}{(l+1)!l!2^{2l+1}}
\end{aligned}$$

Applying Stirling's approximation, we have:

$$\begin{aligned}
\frac{\operatorname{vol}(2l)}{\operatorname{vol}(2l+1)} &= \frac{1}{2R} \frac{\sqrt{2\pi(2l+2)}}{2\pi\sqrt{l(l+1)}} \frac{1}{2^{2l+1}} \left(\frac{2l+2}{e}\right)^{2l+2} \left(\frac{e}{l}\right)^l \left(\frac{e}{l+1}\right)^{(l+1)} \\
&= \frac{1}{2R} \frac{1}{\sqrt{\pi l}} \frac{2}{e} \left(\frac{l+1}{l}\right)^l (l+1) \\
&= \frac{1}{R\sqrt{\pi}} \frac{l+1}{\sqrt{l}} \\
&= \frac{1}{\sqrt{\pi}} \left(1 + \sqrt{\frac{1}{l(l+2)}}\right)
\end{aligned}$$

Returning to the integrand, we can simplify it somewhat:

$$\int_{-R}^{R} x_1^4 \left(1 - \frac{x_1^2}{R^2}\right)^l dx_1 = 2 \int_0^R x_1^4 \left(1 - \frac{x_1^2}{R^2}\right)^l dx_1$$

By explicitly taking the integral (using a computer algebra system), we have:

$$\int_0^R x_1^4 \left(1 - \frac{x_1^2}{R^2}\right)^l dx_1 = \frac{3\sqrt{\pi}(2l+3)^{5/2}\Gamma(l+1)}{8\Gamma(l+7/2)}$$

where Γ here is the usual gamma function. The behavior of this function is as follows:

$$\lim_{l\to\infty} \frac{3\sqrt{\pi}(2l+3)^{5/2}\Gamma(l+1)}{8\Gamma(l+7/2)} = 3\sqrt{\frac{\pi}{2}}$$

Moreover, the function is monotonic increasing for $l > 0$, and takes on the value $56\sqrt{7}/45$ at $l = 2$. Thus, combining these facts with the estimate of the volume ratios, we can see that the fourth moment of a ball is bounded away from the fourth moment of a standard Gaussian by a constant, hence we can take $\eta = \Omega(1)$. Once we have the relevant subspace V, we can project the samples to V and learn in time $T(k, \epsilon)$. The runtime in this case is $T(k, \epsilon) + C_{k,\epsilon}n^2$. □

As a specialization, when the positive examples are determined by a convex subset of the unit ball, $T(k, \epsilon) \leq (k/\epsilon)^{O(k)}$. In a k-dimensional subspace, we can learn a convex subset of the ball by simply taking the convex hull of $(k/\epsilon)^{O(k)}$ random positive points. From the classical approximation theory of convex bodies [109], we obtain an approximation to the true convex body to within relative error ϵ, giving total runtime $(k/\epsilon)^{O(k)} + C_{k,\epsilon}n^2$. This complements [125] which provides a PCA-based algorithm for learning convex bodies when the distribution in the relevant subspace is also Gaussian. In that paper, it is mentioned that standard PCA fails if the full distributions is not a Gaussian.

We now present an example that relies on boundedness – either of the full distribution in the relevant subspace, or the positive distribution. This rather general result uses relatively many moments.

Application 3 (Compact distribution in relevant subspace)**.** *Let $F = F_V F_W$ where F_W is any Gaussian over $n-k$ dimensions. Take $\mathcal{H}$ to be a robust hypothesis class learnable with complexity $T(k, \epsilon)$. Assume that either F_V or $\mathcal{H}$ has its support contained in $B_{g(k)}(0)$.*

Proposition 5.2.19. *The triple $(k, F, \mathcal{H})$ described in Application 3 is $(g(k), \Omega(1))$ moment-learnable with complexity $T(k, \epsilon) + C_{k,\epsilon}n^{O(g(k)^2)}$.*

Proof. Suppose we run **FindBasis** on the full distribution or the positive distribution, whichever is contained in a ball of radius $g(k)$. Consider the relevant subspace. If we fix

some even moment m then we can give explicit bounds on the moments:

$$\mathbb{E}\left((x_t)^m\right) \leq g(k)^m.$$

On the other hand, the even moments of a Gaussian are given by $(m-1)!! = m!/(m/2)!2^{m/2}$ which grows much more rapidly. If we take logarithms on both sides, then we can find $m = m(k)$ such that:

$$m \log(g(k)) \leq \log\left(\frac{m!}{(m/2)!2^{m/2}}\right)$$

Applying Stirling's approximation yields:

$$\begin{aligned}\frac{m}{2}\log(g(k)^2) &\leq m\log(m) - m - \frac{m}{2}\log\left(\frac{m}{2}\right) + \frac{m}{2} - \frac{m}{2}\log(2) \\ &\leq \frac{m}{2}\log(m) - \frac{m}{2}\end{aligned}$$

So if we pick $m = 2g(k)^2$, then the difference in the moments should be $\Omega(1)$. Thus, simply running **FindBasis** on the full distribution will allow us to recover the relevant subspace, at which point we can learn $\mathcal{H}$ in $\mathbb{R}^k$ (doable in time $T(k)$). It remains to prove that we can sample the first $2g(k)^2$ moments of a bounded distribution efficiently: since it is bounded, all moments exist. In particular, if we require $2g(k)^2$ moments, then the $4g(k)^2$ moment is bounded by $g(k)^{4g(k)^2}$. Then by applying Chebyshev's inequality, we see that we need at most $g(k)^{O(g(k)^2)}$ samples in the relevant subspace. The overall runtime for this algorithm is then $T(k,\epsilon) + C_{k,\epsilon}n^{O(g(k)^2)}$. □

5.3 Fully determined ICA

5.3.1 Overview

We commence our study of ICA with the fully determined ICA case: formally, part of this work is subsumed by Section 5.6 on underdetermined ICA, but the basic algorithm and its analysis are substantially simpler than the general case, yet retains the essential elements of our technique – fourier transforms, polynomial anti-concentration and derivative truncation. On the other hand, it does not require the machinery of our tensor decomposition in Section 4.3, and has the advantages of a very attractive interpretation in terms of reweighted PCA and a very fast recursive variant.

Our basic algorithm, unlike most of the literature on ICA that employs moments, we do not require that our underlying random variables s_i differ from a Gaussian at the fourth moment. In fact, our algorithm can deal with differences from being Gaussian at any moment, though the computational and sample complexities are higher when the differences are at higher moments. We will use *cumulants* as a notion of difference from being a Gaussian. The cumulant of random variable x at order r, denoted by $\text{cum}_r(x)$, is the r^{th} moment with some additional subtractions of polynomials of lower moments.

With a slight loss of generality, we assume that A is unitary. If A is not unitary, we can simply make it approximately so by placing the entire sample in isotropic position. Rigorously arguing about this will require an additional error analysis which yields an additional multiplicative $1 - \epsilon$ error; we will omit such details for the sake of clarity. In any case, our algorithm for underdetermined ICA does not (and cannot) make any such assumption. We assume for simplicity and without real loss of generality that $\mathbb{E}(s_j) = 0$ for all j. We can ensure this by working with samples $x^i - \bar{x}$ instead of the original samples x^i (here $\bar{x}$ is the empirical average of the samples). There is a slight loss of generality because using $\bar{x}$ (as opposed to using $\mathbb{E}(x)$) introduces small errors. These errors can be analysed along with the rest of the errors and do not introduce any new difficulties.

Our algorithm is built on the following structural lemma concerning the second derivatives of the log of the Fourier transform of the sample. We will actually employ the *second characteristic function* or *cumulant generating function* given by $\psi(u) = \log(\phi(u))$. Note that both these definitions are with respect to observed random vector x: when x arises from an ICA model $x = As$, we will also define the component-wise characteristic functions with respect to the underlying s_i variables $\phi_i(u_i) = \mathbb{E}\left(e^{iu_i s_i}\right)$ and $\psi_i(u_i) = \log(\phi_i(u_i))$. Note that both these functions are with respect to the underlying random variables s_i and not the observed random variables x_i. For convenience, we shall also write $g_i = \psi_i''$.

Lemma 5.3.1. *Let $x \in \mathbb{R}^n$ be given by an ICA model $x = As$ where $A \in \mathbb{R}^{n \times n}$ is a unitary matrix and $s \in \mathbb{R}^n$ is an independent random vector. Then*

$$D^2\psi = A\text{diag}\left(\psi_i''((A^T u)_i)\right) A^T.$$

Proof. We will compute the derivatives sequentially. For the first derivative we have

$$\frac{\partial \psi}{\partial u_i} = \frac{1}{\phi(u)} \mathbb{E}\left(i(As)_i e^{iu^T As} \right).$$

The second derivative for indices i, j (not necessarily distinct) is

$$\frac{\partial^2 \psi}{\partial u_i \partial u_j} = -\frac{\mathbb{E}\left((As)_i (As_j) e^{iu^T As}\right) \mathbb{E}\left(e^{iu^T As}\right) - \mathbb{E}\left((As)_i e^{iu^T As}\right) \mathbb{E}\left((As)_j e^{iu^T As}\right)}{\mathbb{E}\left(e^{iu^T As}\right)^2}. \quad (19)$$

By independence of the s_i we have

$$\mathbb{E}\left(e^{iu^T As}\right) = \mathbb{E}\left(e^{i(A^T u)^T s}\right) = \prod_{i=1}^{m} \mathbb{E}\left(e^{i(A^T u)_i s_i}\right).$$

Consider the first term in the numerator of 19

$$\begin{aligned}
&\mathbb{E}\left((As)_i (As_j) e^{iu^T As}\right) \mathbb{E}\left(e^{iu^T As}\right) \\
&= \sum_{i'j'} A_{ii'} A_{jj'} \mathbb{E}\left(s_{i'} s_{j'} e^{iu^T As}\right) \mathbb{E}\left(e^{iu^T As}\right) \\
&= \sum_{i'j'} A_{ii'} A_{jj'} \mathbb{E}\left(s_{i'} s_{j'} e^{i(A^T u)^T s}\right) \mathbb{E}\left(e^{i(A^T u)^T s}\right) \\
&= \sum_{i'j'} A_{ii'} A_{jj'} \mathbb{E}\left(s_{i'} s_{j'} e^{i((A^T u)_{i'} s_{i'} + (A^T u)_{j'} s_{j'})}\right) \cdot \prod_{k \neq i', j'}^{m} \mathbb{E}\left(e^{i(A^T u)_k s_k}\right) \cdot \prod_{k=1}^{m} \mathbb{E}\left(e^{i(A^T u)_k s_k}\right).
\end{aligned} \quad (20)$$

If we perform the same calculation on the second term in the numerator of 19

$$\begin{aligned}
&\mathbb{E}\left((As)_i e^{iu^T As}\right) \mathbb{E}\left((As)_j e^{iu^T As}\right) \\
&= \sum_{i'j'} A_{ii'} A_{jj'} \mathbb{E}\left(s_{i'} e^{i(A^T)_{i'} s_{i'}}\right) \cdot \prod_{k \neq i'} \mathbb{E}\left(e^{i(A^T u)_k s_k}\right) \\
&\quad \times \mathbb{E}\left(s_{j'} e^{i(A^T)_{j'} s_{j'}}\right) \prod_{k \neq j'} \mathbb{E}\left(e^{i(A^T u)_k s_k}\right).
\end{aligned} \quad (21)$$

When $i' \neq j'$

$$\mathbb{E}\left(s_{i'} s_{j'} e^{i((A^T u)_{i'} s_{i'} + (A^T u)_{j'} s_{j'})}\right) = \mathbb{E}\left(s_{i'} e^{i(A^T u)_{i'} s_{i'}}\right) \mathbb{E}\left(s_{j'} e^{i(A^T u)_{j'} s_{j'}}\right).$$

Thus, the contributions for $i' \neq j'$ in (20) and (21) exactly net to zero, and the only terms that survive are when $i' = j'$. Now cancelling out the denominator factors corresponding

to other indices gives

$$D^2\psi = A\text{diag}\left(\frac{\mathbb{E}\left(s_k^2 e^{i(A^T u)_k s_k}\right)\mathbb{E}\left(e^{i(A^T u)_k s_k}\right) - \mathbb{E}\left(s_k e^{i(A^T u)_k s_k}\right)^2}{\mathbb{E}\left(e^{i(A^T u)_k s_k}\right)^2}\right) A^T$$
$$= A\text{diag}\left(\psi_k''((A^T u)_k)\right) A^T.$$

□

A more general version applying to higher derivative tensors is proved later in Section 5.6. It is clear from this lemma that the columns of A are simply the eigenvectors of $D^2\psi$, and the associated eigenvalues are $\psi_k''((A^T u)_k)$. Then, so long as the eigenvalues are all different (i.e., no degenerate eigenspace), then we will be able to recover the columns of A up to complex phase factor. Thus, a diagonalisation of this matrix should yield the columns of A. The analysis of this algorithm essentially shows that we can empirically compute this matrix $D^2\psi$ from samples, and that even in the presence of sampling error, that the eigenvalues are sufficiently well-separated and that the eigenvectors are still accurate. The complicating factor here is that this matrix $D^2\psi$ has complex entries and is not Hermitian, thus we must be far more careful about the perturbation theorems that we use.

The Gaussian function plays an important role in harmonic analysis as the eigenfunction of the Fourier transform operator, and we exploit this property to deal with additive Gaussian noise in our model in Section 5.3.5.

5.3.2 Algorithm

Our algorithm computes the eigenvectors of a covariance matrix reweighted according to random Fourier coefficients.

We make the following comments regarding the efficient realisation of this algorithm. The matrix Σ_u in the algorithm is complex and symmetric, and thus is not Hermitian; its eigenvalue decomposition is more complicated than the usual Hermitian/real-symmetric case. It can be computed in one of two ways. One is to compute the SVD of Σ_u (i.e., compute the eigenvalue decomposition of $\Sigma_u \Sigma_u^*$ which is a real symmetric matrix). Alternatively (and this essentially leads to the fast recursive algorithm in the next section), we can exploit the

Fourier PCA(σ)

1. (Isotropy) Get a sample S from the input distribution and use them to find an isotropic transformation B^{-1} with

$$B^2 = \frac{1}{|S|}\sum_{x\in S}(x-\bar{x})(x-\bar{x})^T.$$

2. (Fourier weights) Pick a random vector u from $N(0,\sigma^2 I_n)$. For every x in a new sample S, compute $y = B^{-1}x$, and its Fourier weight

$$w(y) = \frac{e^{iu^T y}}{\sum_{y\in S} e^{iu^T y}}.$$

3. (Reweighted Covariance) Compute the covariance matrix of the points y reweighted by $w(y)$

$$\mu_u = \frac{1}{|S|}\sum_{y\in S} w(y)y \quad \text{and} \quad \Sigma_u = \frac{1}{|S|}\sum_{y\in S} w(y)(y-\mu_u)(y-\mu_u)^T.$$

4. Compute the eigenmatrix V of Σ_u and output BV.

fact that the real and complex parts have the same eigenvectors, and hence by carefully examining the real and imaginary components, we can recover the eigenvectors. We separate $\Sigma_u = \mathrm{Re}\,(\Sigma_u) + i\,\mathrm{Im}\,(\Sigma_u)$ into its real part and imaginary part, and use an SVD on $\mathrm{Re}\,(\Sigma_u)$ to partition its eigenspace into subspaces with close eigenvalues, and then an SVD of $\mathrm{Im}\,(\Sigma_u)$ in each of these subspaces. Both methods need some care to ensure that eigenvalue gaps in the original matrix are preserved, an important aspect of our applications. We complete the algorithm description for ICA by giving a precise method for determining the eigenmatrix V of the reweighted sample covariance matrix Σ_u. This subroutine below translates a gap in the complex eigenvalues of Σ_u into observable gaps in the real part.

1. Write $\Sigma_u = \mathrm{Re}\,(\Sigma_u) + i\,\mathrm{Im}\,(\Sigma_u)$. Note that both the component matrices are real and symmetric.

2. Compute the eigendecomposition of $\mathrm{Re}\,(\Sigma_u) = U\,\mathrm{diag}\,(r_i)\,U^T$.

3. Partition $r_1,\ldots,r_n$ into blocks $R_1,\ldots,R_l$ so that each block contains a subsequence of

eigenvalues and the gap between consecutive blocks is at least ϵ_0, i.e., $\min_{r \in R_j, s \in R_{j+1}} r - s \geq \epsilon_0$. Let U_j be the eigenvectors corresponding to block R_j.

4. For each $1 \leq j \leq l$, compute the eigenvectors of $U_j^T \operatorname{Im}(\Sigma_u) U_j$ and output V as the full set of eigenvectors (their union).

Lemma 5.3.2. *Suppose Σ_u has eigenvalues $\lambda_1, \ldots, \lambda_n$ and $\epsilon = \min_{i \neq j} \min\{\operatorname{Re}(\lambda_i) - \operatorname{Re}(\lambda_j), \operatorname{Im}(\lambda_i) - \operatorname{Im}(\lambda_j)\}$. Then, with $\epsilon_0 = \epsilon/n$, the above algorithm will recover the eigenvectors of Σ_u.*

Proof. The decomposition of the matrix $\operatorname{Re}(\Sigma_u)$, will accurately recover the eigensubspaces for each block (since their eigenvalues are separated). Moreover, for each block $U_j \operatorname{diag}(r_i) U_j^T$, the real eigenvalues r_i are within a range less than ϵ (since each consecutive pair is within $r_i - r_{i+1} < \epsilon/n$). Thus, for each pair $i, i+1$ in this block, we must have a separation of at least ϵ in the imaginary parts of λ_i, λ_{i+1}, by the definition of ϵ. Therefore the eigenvalues of $Q_j = U_j^T \operatorname{Im}(\Sigma_u) U_j$ are separated by at least ϵ and we will recover the original eigenvectors accurately. □

A slight shift of focus – instead of lower bounding all the gaps, we can also just lower bound the largest gap – yields the faster algorithm in the next section.

To perform ICA, we simply apply Fourier PCA to samples from the input distribution. We will show that for a suitable choice of σ and sample size, this will recover the independent components to any desired accuracy. The main challenge in the analysis is showing that the reweighted covariance matrix will have all its eigenvalues spaced apart sufficiently (in the complex plane). This eigenvalue spacing depends on how far the component distributions are from being Gaussian, as measured by cumulants. Any non-Gaussian distribution will have a nonzero cumulant, and in that sense this is a complete method. We will quantify the gaps in terms of the cumulants to get an effective bound on the eigenvalue spacings. The number of samples is chosen to ensure that the gaps remain almost the same, and we can apply eigenvector perturbation theorems Davis-Kahan or Wedin to recover the eigenvectors to the desired accuracy.

Our main theorem in the analysis of this algorithm is as follows:

Theorem 5.3.3. *Let $x \in \mathbb{R}^n$ be given by an ICA model $x = As$ where $A \in \mathbb{R}^{n \times n}$ is unitary and the s_i are independent, $\mathbb{E}\left(s_i^4\right) \leq M_4$ for some constant, and for each s_i there exists a $k_i \leq k$ such that $|\mathrm{cum}_{k_i}(s_i)| \geq \Delta$ (one of the first k cumulants is large) and $\mathbb{E}\left(|s_i|^k\right) \leq M_k$. For any $\epsilon > 0$, with the following setting of σ,*

$$\sigma = \frac{\Delta}{2k!}\left(\frac{\sqrt{2\pi}}{4(k-1)n^2}\right)^k \cdot \frac{1}{(2e)^{k+1} M_k \log(4n)^{k+1}},$$

Fourier PCA *will recover vectors $\{b_1, \ldots, b_n\}$ such that there exists signs $a_i = \pm 1$ satisfying*

$$\|A_i - b_i\| \leq \epsilon$$

with high probability, using $(ckn)^{2k^2+2}(M_k/\Delta)^{2k+2} M_4^2/\epsilon^2$ samples.

Our analysis proceeds via the analysis of the Fourier transform: for a random vector $x \in \mathbb{R}^n$ distributed according to f, the characteristic function is given by the Fourier transform

$$\phi(u) = \mathbb{E}\left(e^{iu^T x}\right) = \int f(x) e^{iu^T x} dx.$$

We favour the Fourier transform or characteristic function over the Laplace transform (or moment generating function) for the simple reason that the Fourier transform always exists, even for very heavy tailed distributions. In particular, the trivial bound $\left|e^{itx}\right| = 1$ means that once we have a moment bound, we can control the Fourier transform uniformly.

Note that the reweighted covariance matrix in our algorithm is precisely the Hessian second derivative matrix $D^2\psi$:

$$\Sigma_u = D^2\psi = \frac{\mathbb{E}\left((x-\mu_u)(x-\mu_u)^T e^{iu^T x}\right)}{\mathbb{E}\left(e^{iu^T x}\right)},$$

where $\mu_u = \mathbb{E}\left(x e^{iu^T x}\right) / \mathbb{E}\left(e^{iu^T x}\right)$.

In case more than one of the variables are (standard) Gaussians, then a quick calculation will verify that $\psi_i''(u_i) = 1$. Thus, in the presence of such variables the eigenvectors corresponding to the eigenvalue 1 are degenerate and we can not resolve between any linear combination of such vectors. Thus, the model is *indeterminate* when some of the underlying

random variables are too gaussian. To deal with this, one typically hypothesizes that the underlying variables s_i are different from Gaussians. One commonly used way is to postulate that for each s_i the fourth moment or cumulant differs from that of a Gaussian. We weaken this assumption, and only require that *some* moment is different from a Gaussian.

5.3.3 Eigenvalue spacings

To obtain a robust algorithm, we rely on the eigenvalues of $D^2\psi$ being adequately spaced (so that the error arising from sampling does not mix the eigenspaces, hence columns of A). Thus, we inject some randomness by picking a random Fourier coefficient, and hope that the $g_i(u_i)$ are sufficiently anti-concentrated. To this end, we will truncate the Taylor series of g_i to k^{th} order, where the k^{th} cumulant is one that differs from a gaussian substantially. The resulting degree k polynomial will give us the spacings of the eigenvalues via polynomial anti-concentration estimates in Section 3.1, and we will control the remaining terms from order $k+1$ and higher by derivative estimates. Notably, the further that s_i is from being a gaussian (in cumulant terms), the stronger anti-concentration. We will pick the random Fourier coefficient u according to a Gaussian $N(0, \sigma^2 I_n)$ and we will show that with high probability for all pairs i, j we have

$$\left|g_i((A^T u)_i) - g_j((A^T u)_j)\right| \geq \delta.$$

Critical to our analysis is the fact that $(A^T u)_i$ and $(A^T u)_j$ are both independent Gaussians since the columns of A are independent by our assumption of isotropic position (Section 5.3.3). We then go onto compute the sample complexity required to maintain these gaps in Section 5.3.4 and conclude with the proof of correctness for our algorithm in the last section.

To begin: since we are taking the derivatives of $\log(\phi)$, we will establish that this is well-defined (i.e., that $1/\phi$ is in fact bounded). We will use this fact repeatedly in the succeeding section.

Lemma 5.3.4 (Nonvanishing of $\phi(t)$)**.** *Let s be a real-valued random vector in $\mathbb{R}^m$ with independent components and $\mathbb{E}(s) = 0$. Also let $\mathbb{E}(|s_j|)$ and $\mathbb{E}\left(\left|s_j^2\right|\right)$ exist and $\mathbb{E}\left(\left|s_j^2\right|\right) \leq$*

M_2 for all j for $M_2 > 0$. Then for $t \in \mathbb{R}^m$ with $\|t\|_2 \leq \frac{1}{2\sqrt{M_2}}$ the characteristic function $\phi(\cdot)$ of s satisfies $|\phi(t)| \geq 3/4$.

Proof. Using Taylor's theorem 5.3.5 for $\cos y$ and $\sin y$ gives

$$e^{iy} = \cos y + i \sin y = 1 + iy - \frac{(iy)^2}{2!}[\cos(\theta_1 y) + i \sin(\theta_2 y)],$$

for $y, \theta_1, \theta_2 \in \mathbb{R}$ with $|\theta_1| \leq 1, |\theta_2| \leq 1$. Applying this to $y = t^T s$, taking expectation over s, and using the assumption of zero means on the s_i we get

$$\mathbb{E}\left(e^{it^T s}\right) = 1 - \mathbb{E}\left(\frac{(it^T s)^2}{2}[\cos(\theta_1 y) + i \sin(\theta_2 y)]\right),$$

which using the independence of the components of s and the zero means assumption gives

$$\begin{aligned}
\left|\mathbb{E}\left(e^{it^T s}\right) - 1\right| = |\phi(t) - 1| &\leq \frac{1}{2}\mathbb{E}\left((t^T s)^2 \left|\cos(\theta_1 y) + i \sin(\theta_2 y)\right|\right) \\
&\leq \mathbb{E}\left((t^T s)^2\right) \\
&= \sum_j t_j^2 \mathbb{E}\left(s_j^2\right) \\
&\leq R_2 \|t\|_2^2 \\
&\leq 1/4.
\end{aligned}$$

□

Next, let us commence our program of polynomial truncation and expand out the function $g_i = \psi_i''$ as a Taylor series with error estimate:

Theorem 5.3.5 (Taylor's theorem with remainder)**.** *Let $f : \mathbb{R} \to \mathbb{R}$ be a C^n continuous function over some interval I. Let $a, b \in I$, then*

$$f(b) = \sum_{k=1}^{n-1} \frac{f^{(k)}(a)}{k!}(b-a)^k + \frac{f^{(n)}(\xi)}{n!}(b-a)^n,$$

for some $\xi \in [a, b]$.

To this end, we write

$$g_i(u_i) = p_i(u_i) + \frac{g^{(k)}(\xi)}{k!}u_i^k, \tag{22}$$

where $\xi \in [0, u_i]$ and p_i is a polynomial of degree $(k-1)$.

To bound the error term in (22), we observe that it suffices to bound $[\log(\phi_i)]^{(k)}(u_i)$ using the following lemma.

Lemma 5.3.6. *Let $x \in \mathbb{R}$ be a random variable with finite k absolute moments, and let $\phi(u)$ be the associated characteristic function. Then*

$$\left|[\log(\phi)]^{(k)}(u)\right| \leq \frac{2^{k-1}(k-1)!\,\mathbb{E}\left(|x|^k\right)}{|\phi(u)|^k}.$$

Proof. We will compute the derivatives of $\psi(u) = \log \phi(u)$ as follows: we proceed recursively with $\psi'(u) = \phi'(u)/\phi(u)$ as our base case. Let $\psi^{(d)}$ be given by the ratio of two functions, a numerator function $N(u;d)$ and a denominator function $D(u;d)$, with no common factors and $N(u;d)$ is the sum of terms of the form $\prod_{j=1}^{d} \phi^{(i_j)}(u)$ where the coefficient of each term is ± 1. Some useful properties of functions $N(u;d), D(u;d)$ are summarized in the following claim.

Claim 5.3.7. *For $d \geq 1$, functions $N(u;d)$ and $D(u;d)$ satisfy*

1. *$D(u;d) = \phi(u)^d$.*
2. *For each term of $N(u;d)$, $\sum_{j=1}^{d} i_j \leq d$.*
3. *For each term of $N(u;d)$, the total number of factors of ϕ and its derivatives is at most d.*
4. *For $d \geq 1$, there are at most $2^{d-1}(d-1)!$ terms in $N(u;d)$.*

Proof. We will prove all these via induction over d. Clearly these are all true for the base case $d = 1$. Assume that all four facts are true for some d, we will now examine the case for $d+1$.

Writing $\psi^{(d+1)}(u)$ as the derivative of $\psi^{(d)}(u) = N(u;d)/D(u;d)$ and canceling common

factors gives

$$\begin{aligned}\psi^{(d+1)}(u) =& \frac{N'(u;d)D(u;d) - N(u;d)D'(u;d)}{D(u;d)^2} \\ =& \frac{N'(u;d)\phi(u)^d - N(u;d)d\phi(u)^{d-1}\phi'(u)}{\phi(u)^{2d}} \\ =& \frac{N'(u;d)\phi(u) - d\phi'(u)N(u;d)}{\phi(u)^{d+1}}. \end{aligned} \tag{23}$$

Observing that there is always a term in $N(u;d) = \phi'(u)^d$, we can not cancel any further factors of $\phi(u)$. Hence $D(u;d) = \phi(u)^d$, proving the first part of the claim.

The second and third parts of the claim follow immediately from the final expression for $\psi^{(d+1)}(u)$ above and our inductive hypothesis.

To prove the fourth part, let $T(d)$ denote the total number of terms in $N(u;d)$, then by part 3 and the expansion in (23), we have $T(d+1) \leq dT(d) + dT(d) \leq 2dT(d)$. From this $T(d+1) \leq 2^d d!$ follows immediately. □

Returning to the proof of Lemma 5.3.6, for $d \leq k$ we observe that

$$\left|\phi^{(d)}(u)\right| = \left|\mathbb{E}\left((ix)^d e^{iu^T x}\right)\right| \leq \mathbb{E}\left(\left|(ix)^d e^{iu^T x}\right|\right) \leq \mathbb{E}\left(|x|^d\right).$$

Thus, for each term of $N(u;d)$:

$$\left|\prod_{j=1}^{d} \phi^{(i_j)}(u)\right| \leq \prod_{j=1}^{d} \left|\phi^{(i_j)}(u)\right| \leq \prod_{j=1}^{d} \mathbb{E}\left(|x|^{i_j}\right) \overset{\text{Fact3.1.1}}{\leq} \mathbb{E}\left(|x|^{\sum_{j=1}^{d} i_j}\right) \leq \mathbb{E}\left(|x|^d\right).$$

Combining Claim 5.3.7 with the previous equation, and noticing that we never need to consider absolute moments of order higher than k (which are guaranteed to exist by our hypothesis), gives the desired conclusion. □

To conclude this calculation of truncation error, observe that if the distribution of $x \in \mathbb{R}$ is isotropic then for $u \in (-1,1)$ we have

$$\phi_x(u) = \phi_x(0) + \phi_x'(0)u + \frac{\phi_x''(\xi)}{2}u^2,$$

where $\xi \in [0,u]$. We have $\phi_x(0) = 1$, $\phi_x'(0) = \mathbb{E}(x) = 0$ and $|\phi_x''(\xi)| \leq \mathbb{E}\left(|x|^2\right) = 1$ by the isotropic position assumption. Thus, for $u \in [-1/4, 1/4]$, Lemma 5.3.6 gives us

$$\left|[\log(\phi_x)]^{(k)}(u)\right| \leq \mathbb{E}\left(|x|^k\right) k^k. \tag{24}$$

We are now ready to compute the spacings of the diagonal matrix.

Theorem 5.3.8. *Let $s \in \mathbb{R}^n$ be a random vector with indpendent components. For $t \in \mathbb{R}^n$, let $\psi(t) = \log \mathbb{E}\left(e^{it^T s}\right)$ be the second characteristic function of s. Suppose we are given the following data and conditions:*

1. *Integer $k > 2$ such that $\mathbb{E}\left(|s_i|^k\right)$ exists for all $i \in [n]$.*
2. *$\Delta > 0$ such that for each $i \in [n]$, there exists $2 < k_i < k$ such that $|\mathrm{cum}_{k_i}(s_i)| \geq \Delta$.*
3. *$M_2 > 0$ such that $\mathbb{E}\left(s_i^2\right) \leq M_2$ for $i \in [n]$.*
4. *$M_k > 0$ such that $\mathbb{E}\left(|s_i|^k\right) \leq M_k$ for $i \in [n]$.*
5. *$g_i(t_i) := \frac{\partial^2 \psi(t)}{\partial t_i^2}$.*
6. *$\tau \sim N(0, \sigma^2 I_n)$ where $\sigma = \min(1, \frac{1}{2\sqrt{2M_2 \log 1/q}}, \sigma')$ and*

$$\sigma' = \left(\frac{3}{8}\right)^{k+1} \cdot \frac{k-1}{k!} \cdot \left(\frac{\sqrt{2\pi}}{4k}\right)^{k-2} \cdot \frac{q^{k-2}}{(\sqrt{2\log(1/q)})^{k-1}} \cdot \frac{\Delta}{M_k}, \tag{25}$$

   and $0 < q \leq 1/3$.

Then with probability at least $1 - n^2 q$, for all distinct i, j we have

$$|g_i(\tau_i) - g_j(\tau_j)| \geq \frac{\Delta}{2(k-2)!} \left(\frac{\sqrt{2\pi}\sigma q}{4k}\right)^{k-2}.$$

Proof. We will argue about the spacing $|g_1(\tau_1) - g_2(\tau_2)|$, and then use the union bound to get that none of the spacings is small with high probability. Since s_1 has first k moments, we can apply Taylor's theorem with remainder (Actually one needs more care as that theorem was stated for functions of type $\mathbb{R} \to \mathbb{R}$, whereas our function here is of type $\mathbb{R} \to \mathbb{C}$. To this end, we can consider the real and imaginary parts of the function separately and apply Theorem 5.3.5 to each part; we omit the details.) Applying Theorem 5.3.5 gives

$$g_1(t_1) = -\sum_{l=2}^{k_1} \mathrm{cum}_l(s_1) \frac{(it_1)^{l-2}}{(l-2)!} + R_1(t_1) \frac{(it_1)^{k_1-1}}{(k_1-1)!}.$$

Truncating g_1 after the degree $(k_1 - 2)$ term yields a polynomial $p_1(t_1)$. Denote the truncation error by $\rho_1(t_1)$. Then, fixing t_2 arbitrarily and setting $z = g_2(t_2)$ for brevity, we have

$$\begin{aligned} |g_1(t_1) - g_2(t_2)| &= |p_1(t_1) + \rho_1(t_1) - z| \\ &\geq |p_1(t_1) - z| - |\rho_1(t_1)| \,. \end{aligned}$$

We will show that $|p_1(t_1) - z|$ is likely to be large and $|\rho_1(t_1)|$ is likely to be small. Noting that $\frac{(k_1-2)!}{i^{k_1-2}\mathrm{cum}_{k_1}(s_1)} p_1(t_1)$ is monic of degree $k_1 - 2$ (but with coefficients from $\mathbb{C}$), we apply our anti-concentration result in Theorem 3.1.5. Again, although that theorem was proven for polynomials with real coefficients, its application to the present situation is easily seen to go through without altering the bound by considering the real and imaginary parts separately. In the following, the probability is for $t_1 \sim N(0, \sigma^2)$.

$$\mathsf{Pr}\left(|p_1(t_1) - z| \leq \epsilon_1\right) \leq \frac{4(k_1-2)}{\sigma\sqrt{2\pi}} \left(\frac{\epsilon_1 (k_1-2)!}{|\mathrm{cum}_{k_1}(s_1)|}\right)^{1/(k_1-2)} \leq \frac{4k_1}{\sigma\sqrt{2\pi}} \left(\frac{\epsilon_1 (k_1-2)!}{\Delta}\right)^{1/(k_1-2)}.$$

Setting

$$\epsilon_1 := \frac{\Delta}{(k_1-2)!} \left(\frac{\sqrt{2\pi}\sigma q}{4k_1}\right)^{(k_1-2)} \leq \frac{\Delta}{(k-2)!} \left(\frac{\sqrt{2\pi}\sigma q}{4k}\right)^{(k-2)} \tag{26}$$

we have

$$\mathsf{Pr}\left(|p_1(t_1) - z| \leq \epsilon\right) \leq q.$$

Next we bound the truncation error and show that $|\rho_1(t_1)| \leq \epsilon/2$ with probability at least $1 - \frac{q}{\sqrt{\pi \log 1/q}}$. Applying Lemma 5.3.6, the error introduced is

$$|\rho_1(t_1)| \leq \frac{k_1!\, 2^{k_1} \mathbb{E}\left(|s_1|^{k_1+1}\right)}{|\phi_1(t_1)|^{k_1+1}} \cdot \frac{t_1^{k_1-1}}{(k_1-1)!}.$$

We now lower bound the probability that $|t_1|$ is small when $t_1 \sim N(0, \sigma^2)$:

$$\mathsf{Pr}\left(|t_1| \leq \sigma\sqrt{2\log 1/q}\right) \geq 1 - \frac{q}{\sqrt{\pi \log 1/q}}.$$

The computation above used Claim 3.1.2.

Thus with probability at least $1 - \frac{q}{\sqrt{\pi \log 1/q}}$ we have

$$|\rho_1(t_1)| \leq \frac{k_1!\, 2^{k_1} M_k}{(3/4)^{k_1+1}(k_1-1)!} \cdot (\sigma\sqrt{2\log 1/q})^{k_1-1}, \tag{27}$$

here we used that by our choice of σ we have $\sigma\sqrt{2\log 1/q} \leq \frac{1}{2\sqrt{M_2}}$, hence Lemma 5.3.4 gives that $|\phi(t_1)| \geq 3/4$.

Now for $|t_1| \leq \sigma\sqrt{2\log 1/q}$ we want

$$|\rho_1(t_1)| \leq \epsilon_1/2.$$

This is seen to be true by plugging in the value of ϵ_1 from (26) and the bound on $\rho_1(t_1)$ from (27) and our choice of σ.

Thus we have proven that $|g_1(t_1) - g_2(t_2)| \geq \epsilon/2$ with probability at least $1 - (q + \frac{q}{\sqrt{\pi \log 1/q}}) \geq 1 - 2q$ (using $q \in (0, 1/3]$). Now applying the union bound over all pairs we get the required bound. □

5.3.4 Proof of the main theorem

In this section, we bound the sample complexity of the algorithm and complte the proof of the main theorem. First we will show how many samples are necessary to estimate accurately the desired Fourier transforms $\mathbb{E}\left(e^{iu^T x}\right)$, $\mathbb{E}\left(xe^{iu^T x}\right)$ and $\mathbb{E}\left(xx^T e^{iu^T x}\right)$.

Lemma 5.3.9. *Let $x \in \mathbb{R}^n$ be a random vector. Fix $\epsilon > 0$ and a vector $t \in \mathbb{R}^n$. Let $x^{(j)}$ be i.i.d. samples drawn according to x then*

$$\left| \frac{1}{m} \sum_{j=1}^{m} e^{iu^T x^{(j)}} - \mathbb{E}\left(e^{iu^T x}\right) \right| \leq \epsilon,$$

with probability at least $1 - 4e^{-m\epsilon^2/2}$.

Proof. Note that the random variables $e^{iu^T x}$ are bounded in magnitude by 1. We separate out the real and imaginary components of $e^{iu^T x}$ and separately apply the Chernoff inequality. □

In the most general setting, all we can do is bound the variance of our sample covariance matrix, and this will give a polynomial bound on the sample complexity.

Lemma 5.3.10. *Suppose that the random vector $x \in \mathbb{R}^n$ is drawn from an isotropic distribution F. Then*

$$\mathrm{Var}(x_j e^{iu^T x}) \leq 1 \text{ for } 1 \leq j \leq n,$$

$$\mathrm{Var}(x_j^2 e^{iu^T x}) \leq \mathbb{E}\left(x_j^4\right),$$

$$\mathrm{Var}(x_i x_j e^{iu^T x}) \leq 1 \text{ for } i \neq j.$$

Proof.

$$\mathrm{Var}(x_j e^{iu^T x}) = \mathbb{E}\left(x_j^2\right) - \left|\mathbb{E}\left(x_j e^{iu^T x}\right)\right|^2 \leq 1.$$

The other parts are similar, with the last inequality using isotropy. □

We can combine these concentration results for the Fourier derivatives to obtain the final sample complexity bound. Recall from (24) that we have in the interval $u \in [-1/4, 1/4]$

$$|g(u)| \leq \mathbb{E}\left(|x|^2\right) k^k \leq 16$$

We can now give the sample complexity of the algorithm.

Corollary 5.3.11. *Let $x = As$ be an ICA model where $A \in \mathbb{R}^{n \times n}$ is a unitary matrix. Suppose that the random vector $s \in \mathbb{R}^n$ is drawn from an isotropic distribution, and that for each s_i, we have $\mathbb{E}\left(s_i^4\right) \leq M$. Fix $\epsilon > 0$ and a vector $u \in \mathbb{R}^n$ where $\|u\| \leq 1/4$. Let $\hat{\Sigma}_u$ be the matrix estimated from m independent samples of $x^i = As^i$, then*

$$\left\|\hat{\Sigma}_u - \Sigma_u\right\|_F \leq \epsilon$$

with probability at least $1 - 1/n$ for $m \geq \mathrm{poly}(n, M)/\epsilon^2$.

Proof. Apply Chebyshev's inequality along with the variance bounds. Since the Frobenius norm is unitarily invariant, we can consider the error in the basis corresponding to s. In this basis:

$$\begin{aligned}
&\left\|\mathbb{E}\left(e^{iu^T s}(s-\tilde{\mu})(s-\tilde{\mu})^T - \tilde{\Sigma}_u\right)\right\| \\
&\leq \left\|\mathbb{E}\left(ss^T e^{iu^T s}\right) - \sum_{i=1}^m (s^i)(s^i)^T e^{iu^T s^i}\right\| + 2\left\|\mathbb{E}\left(s\tilde{\mu}^T e^{iu^T s}\right) - \sum_{i=1}^m x\hat{\tilde{\mu}}^T e^{iu^T s}\right\| \\
&\quad + \left\|\mathbb{E}\left(\tilde{\mu}\tilde{\mu}^T\right) - \hat{\tilde{\mu}}\hat{\tilde{\mu}}^T\right\| \left|\mathbb{E}\left(e^{iu^T s}\right)\right| \\
&\leq \epsilon
\end{aligned}$$

where the last bound is derived by apportioning $\epsilon/5$ error to each term. Finally, we conclude by noting that by our choice of t, we have $\left|\mathbb{E}\left(e^{iu^T x}\right)\right| \geq 29/32$, and the multiplicative error due to the scaling by $1/\mathbb{E}\left(e^{iu^T x}\right)$ is lower order in comparison to ϵ. □

For more structured distributions, e.g., logconcave distributions, or more generally distributions with subexpoential tails, much sharper bounds are known on the sample complexity of covariance estimation, see e.g., [114, **?**, 121, 2].

We can now finish the proof of the main theorem.

Proof of Theorem 5.3.3. In the exact case, the diagonal entries are given by $g_i((A^T u)_i)$. Since A is orthonormal, for any pair $(A^T u)_i = A_i^T u$ and $(A^T u)_j = A_j^T u$ have orthogonal A_i and A_j, hence the arguments of g_i and g_j are independent Gaussians and Theorem 5.3.8 gives us the eigenvalue spacings of Σ_u to be used in Lemma 3.2.5.

In particular, the spacings are at least $\xi = \frac{\Delta}{2k!}\left(\frac{\sqrt{2\pi}\sigma}{4(k-1)^{n^2}}\right)^k$. Thus, with desired accuracy ϵ in Lemma 3.2.5, then we require the sampling error (in operator norm, which we upper bound using Frobenius norm) to be $\|E\|_F \leq \epsilon\xi/(\xi+\epsilon)$. We can then substitute this directly into Corollary 5.3.11 which gives the sample complexity. □

5.3.5 Gaussian noise

The Gaussian function has several nice properties with respect to the Fourier transform, and we can exploit these to cancel out independent Gaussian noise in the problems that we study. To deal with Gaussian noise, when the observed signal $x = As + \eta$ where η is from an unknown Gaussian $N(\mu_\eta, R_\eta)$ which is independent of s, we can use the following modified algorithm.

1. Pick two different random Gaussian vectors u, v.
2. Compute $\Sigma = \Sigma_0, \Sigma_u$ and Σ_v as in the previous algorithm.
3. Output the eigenvectors of $(\Sigma_u - \Sigma)(\Sigma_v - \Sigma)^{-1}$.

Theorem 5.3.12. *Let $x \in \mathbb{R}^n$ be given be a noisy independent components model $x = As+\eta$, where $A \in \mathbb{R}^{n\times n}$ is a full rank matrix, and the noise vector η has a Gaussian distribution. With sufficiently many samples, the modified algorithm outputs A.*

Proof. When $x = As + \eta$, the function $\psi(u) = \log\left(\mathbb{E}\left(e^{iu^T x}\right)\right)$ can be written as

$$\psi(u) = \log\left(\mathbb{E}\left(e^{iu^T x}\right)\right) + \log\left(\mathbb{E}\left(e^{iu^T \eta}\right)\right)$$

Therefore,

$$\begin{aligned} D^2\psi_u &= A\text{diag}\left(\psi_i''(A_i^T u)\right) A^T + \frac{\mathbb{E}\left(e^{iu^T\eta}(\eta-\mu_\eta)(\eta-\mu_\eta)^T\right)}{\mathbb{E}\left(e^{iu^T\eta}\right)} \\ &= A\text{diag}\left(\psi_i''(A_i^T u)\right) A^T + \mathbb{E}\left((\eta-\mu_\eta)(\eta-\mu_\eta)^T\right) \\ &= A\text{diag}\left(\psi_i''(A_i^T u)\right) A^T + R_\eta \end{aligned}$$

where $\eta \sim N(\mu_\eta, R_\eta)$. Therefore,

$$\Sigma_u - \Sigma = A(D_u - D)A^T$$

with D being the covariance matrix of s and

$$(\Sigma_u - \Sigma)(\Sigma_v - \Sigma)^{-1} = A(D_u - D)(D_v - D)^{-1}A^{-1}.$$

The eigenvectors of the above matrix are the columns of A. □

For a complete robustness analysis, one needs to control the spectral perturbations of the matrix $A(D_u - D)(D_v - D)^{-1}A^{-1}$ under sampling error. We omit this proof, but note that it follows easily using the techniques we develop for underdetermined ICA.

5.4 Fast recursive partitioning algorithm

We now give an algorithm for the traditional case of fully-determined ICA, when all the latent variables s_i differ from Gaussian in the fourth moment by at least Δ. The algorithm we give is extremely efficient in terms of sample complexity, and we explore this behaviour in this section. The new algorithm is based on the same structural properties as the algorithm in the previous section; the major insight is that instead of having to space all the eigenvalues

along the real line, it simply suffices if there is a single large gap between some adjacent pair of eigenvalues on the real line. In this case, we can simply split the eigenvectors into two sets according to where their eigenvalues fall relative to this large gap. We can then simply project the samples onto the two subspaces spanned by these sets and proceed recursively. The best fourth moment tensor based algorithms all must construct the fourth moment tensor from samples (either explicitly or implicitly); this leads to a lower bound of $O(n^2)$ samples [67]. Our algorithm uses $\tilde{O}(n)$ samples.

Recursive Fourier PCA(σ, Projection matrix $P \in \mathbb{R}^{n \times k}$)

1. (Termination check) If $k = 1$, return P.
2. (Projection) Project all samples by multiplying by P^T to projected samples S.
3. (Isotropy) Find an isotropic transformation B^{-1} with
$$B^2 = \frac{1}{|S|} \sum_{x \in S} (x - \bar{x})(x - \bar{x})^T.$$
4. (Fourier weights) Pick a random vector u from $N(0, \sigma^2 I_k)$. For every x in a new sample S, compute $y = B^{-1}x$, and its Fourier weight
$$w(y) = \frac{e^{iu^T y}}{\sum_{y \in S} e^{iu^T y}}.$$
5. (Reweighted Covariance) Compute the covariance matrix of the points y reweighted by $w(y)$
$$\mu_u = \frac{1}{|S|} \sum_{y \in S} w(y) y \quad \text{and} \quad \Sigma_u = \frac{1}{|S|} \sum_{y \in S} w(y)(y - \mu_u)(y - \mu_u)^T.$$
6. Compute the spectral decomposition $\{\lambda_i\}$, $\{v_i\}$ of $\mathrm{Re}(\Sigma_u)$.
7. (Eigenvalue gaps) Find the largest gap $\lambda_{i+1} - \lambda_i$. If the gap is too small, pick a different random vector u. Partition the eigenvectors into $V_1 = \{v_1, \ldots, v_i\}$ and $V_2 = \{v_{i+1}, \ldots, v_k\}$.
8. Solve the subproblems $W_1 =$ Recursive FPCA(σ, PV_1) and $W_2 =$ Recursive FPCA(σ, PV_2)
9. Return $[W_1 \quad W_2]$.

5.4.1 Analysis

The analysis of the recursive FPCA algorithm directly uses the $\sin(\theta)$ of Davis and Kahan [51]. Roughly speaking, the largest eigenvalue gap controls the magnitude of the error in each subspace V_1 and V_2 in the algorithm, each recursive step subsequently accumulates error accordingly, and we have to solve a non-linear recurrence to bound the total error.

Theorem 5.4.1. *Let $x \in \mathbb{R}^n$ be given by an ICA model $x = As$ where $A \in \mathbb{R}^{n \times n}$ is unitary, the s_i are independent, $\|s\| \leq K\sqrt{n}$ almost surely, and for each s_i, $|\mathrm{cum}_4(s_i)| \geq \Delta$. For any $\epsilon > 0$, with the following setting of $0 < \sigma < \Delta/100\sqrt{2\log(n)}\max_i \mathbb{E}\left(|s_i|^5\right)^2$ **Recursive FPCA** will recover vectors $\{b_1, \ldots, b_n\}$ such that there exists signs $a_i = \pm 1$ satisfying*

$$\|A_i - b_i\| \leq \epsilon$$

with high probability, using $O(K^2 n \log(n)^7/\Delta^6\epsilon^2)$ samples.

Proof. The proof develops similarly to the proof of Theorem 5.3.8. First, we shall prove that at the top level of the recursive algorithm, that there exists at least one large gap in the set $\{g_i(\tau_i)\}$. To this end, we apply Taylor's Theorem as before, which gives:

$$g_1(t_1) = 1 + \mathrm{cum}_1(s_1)(it_1) + \mathrm{cum}_2(s_1)\frac{(it_1)^2}{2!} + R_1(t_1)\frac{(it_1)^3}{3!}.$$

Now, when we take the real part of the matrix in step 6 of the algorithm, we can discard the pure imaginary term arising from the first cumulant (note that we must retain the error term as we do not know *a priori* whether the error derivative term has a complex component or not). Now, truncating after the second order terms, this gives a family of polynomials $p_j(t_j) = 1 - \mathrm{cum}_2(s_j)t^2/2$. We can now apply Theorem 3.1.9 that shows that with probability $1/2000\log(n)^2$, that the maximum gap in the set $\{p_j(t_j)\}$ is at least $\Delta\sigma^2\log(2)/10\log(n)$. Thus with $8000\log(n)^3$ different u, with probability at least $1/n^2$ we will see a gap of such size. Next, note that by our choice of σ, that with probability at least $1 - 1/n^2$, the remainder term is bounded $\left|R_j(t_j)(it_j)^3\right| \leq \Delta\sigma^2\log(2)/40\log(n)$, thus overall we have that with high probability the set of $\{g_j(t_j)\}$ have a maximum gap of size at least $\Delta\sigma^2\log(2)/20\log(n)$.

Next, partition the eigenvectors according to which side of the maximum gap they fall on, let V and $V^{\perp}$ denote these sets respectively. We bound the error in terms of the $\sin(\theta)$ error of Theorem 3.2.3. Suppose that in each iteration, we take enough samples so that the empirical version of $D^2 \log(\phi)_u$ is within ϵ' of the true one. Then applying the $\sin(\theta)$ theorem yields that the for the subspaces spanned by V and $W = V^{\perp}$, that there exists a partition of the columns of A (which we may take, without loss of generality, to be ordered appropriately) such that:

$$\|\sin(\theta(V, \{A_1, \ldots, A_k\}))\| \leq \frac{\epsilon'}{\frac{\Delta\sigma^2 \log(2)}{20 \log(n)}}$$

Now consider the call of Recursive FPCA on the subspace V of dimension k. In this subspace, we can write the Hessian matrix as:

$$\begin{aligned} D^2 \log(\phi)_u = &(V^T[A_1, \ldots, A_k])\text{diag}\left(\lambda_1, \ldots, \lambda_k\right)\left(V^T[A_1, \ldots, A_k]\right)^T \\ &+ (V^T[A_{k+1}, \ldots, A_n])\text{diag}\left(\lambda_{k+1}, \ldots, \lambda_n\right)\left(V^T[A_{k+1}, \ldots, A_n]\right)^T \end{aligned}$$

Note that by definition, we have that $\sin(\theta) = V^T[A_{k+1}, \ldots, A_n]$, thus the second term is upper bounded by $(20\epsilon' \log(n)/\log(2)\Delta)^2$; we must also add the sampling error from the second iteration (say another ϵ'). In particular, suppose that we write the recurrence for the overall error E_k at a recursive call at depth k, then:

$$E_k = \epsilon' + \left(\frac{E_{k-1} \log(n)}{c\Delta}\right)^2$$

For small ϵ' (to be determined later), we can simply solve the following recurrence to bound the total error by $2\epsilon'$.

Claim 5.4.2. *Fix $a, b > 0$ where $4/b^2 \leq 1$, and define the recurrence $y_{i+1} = a + (y_i/b)^2$ and $y_0 = 0$, then $y_i \leq 2a$ for all i.*

Proof. We proceed via induction. Clearly this is true for $i = 0$. Now suppose that it is true for $i \leq k$, then:

$$y_{i+1} = a + (y_i/b)^2 \leq a + 4a/b^2 \leq 2a$$

as required. □

In the terminal nodes of the recurrence, this gets blown up to $40\log(n)\epsilon'/\Delta\sigma^2\log(2)$; in this iteration the output error will give the overlap between the output vectors and those of A. Thus, setting $\epsilon' = \epsilon\Delta^3/2\log(n)^2\max_i \mathbb{E}\left(|s_i|^5\right)^2$ suffices to give total error ϵ.

For the sample complexity, we simply have to take enough samples so that for $8000log(n)^3$ different instantiations of the Fourier derivative matrix, that the spectral norm error is within ϵ' with high probability. To this end, we can apply Theorem 1.2 from [130] by first splitting the real and imaginary part of the second derivative matrix, and then splitting these further into positive and negative parts.

Theorem 5.4.3 ([130]). *Consider a random vector $x \in \mathbb{R}^n$ with covariance Σ, such that $\|x\| \leq \sqrt{m}$ almost surely. Let $\epsilon \in (0,1)$ and $t \geq 1$, then with probability at least $1 - 1/n^{t^2}$, if $N \geq C(t/\epsilon)^2 \|\Sigma\|^{-1} m\log(n)$, then $\|\Sigma_N - \Sigma\| \leq \epsilon\|\Sigma\|$.*

In particular, it suffices to estimate three matrix valued random variables $\mathbb{E}\left(xx^T\exp(iu^Tx)\right)$, $\mathbb{E}\left(x\exp(iu^Tx)\right)$ and $\mathbb{E}\left(\exp(iu^Tx)\right)$. The latter two are easy to estimate using $O(n)$ samples by applying Lemma 5.3.10. Thus, it suffices for us to show that we can estimate the second order term $\mathbb{E}\left(xx^T\exp(iu^Tx)\right)$ using only a linear number of samples. To this end, let us rewrite this term into four easily estimable parts:

$$\begin{aligned}
\mathbb{E}\left(xx^T\exp(iu^Tx)\right) &= \mathbb{E}\left(xx^T\cos(u^Tx)\right) + i\mathbb{E}\left(xx^T\sin(u^Tx)\right) \\
&= \mathbb{E}\left(xx^T\mathbb{1}_{\cos(u^Tx)\geq 0}\cos(u^Tx)\right) + \mathbb{E}\left(xx^T\mathbb{1}_{\cos(u^Tx)<0}\cos(u^Tx)\right) \\
&\quad + i\mathbb{E}\left(xx^T\mathbb{1}_{\sin(u^Tx)\geq 0}\sin(u^Tx)\right) + i\mathbb{E}\left(xx^T\mathbb{1}_{\sin(u^Tx)<0}\sin(u^Tx)\right) \\
&= \mathbb{E}\left(xx^T\mathbb{1}_{\cos(u^Tx)\geq 0}\cos(u^Tx)\right) - \mathbb{E}\left(xx^T\mathbb{1}_{\cos(u^Tx)<0}\left|\cos(u^Tx)\right|\right) \\
&\quad + i\mathbb{E}\left(xx^T\mathbb{1}_{\sin(u^Tx)\geq 0}\sin(u^Tx)\right) - i\mathbb{E}\left(xx^T\mathbb{1}_{\sin(u^Tx)<0}\left|\sin(u^Tx)\right|\right)
\end{aligned}$$

Let us estimate these four quantities using independent samples – if each one is within $\epsilon/4$, then certainly we can estimate $\mathbb{E}\left(xx^T\exp(iu^Tx)\right)$ to within ϵ in the spectral norm. Consider, for example, the first term $xx^T\mathbb{1}_{\cos u^Tx\geq 0}\cos(u^Tx)$, then we can define the random vector $y = x\mathbb{1}_{\cos(u^Tx)\geq 0}\sqrt{\cos(u^Tx)}$. Then it is clear that:

$$yy^T = xx^T\mathbb{1}_{\cos u^Tx\geq 0}\cos(u^Tx)$$

In particular, observe that $0 \leq \mathbb{E}\left((u^T y)^2\right) \leq \mathbb{E}\left((u^T x)^2\right) \leq 1$ for all unit vectors u. Thus, we must have that the eigenvalues of $\mathbb{E}\left(yy^T\right)$ are all bounded by 1. Note also, that $\|y\| \leq \sqrt{m}$ if this is in fact the case for x as well. Now, we apply Theorem 1.2 from [130] to y: by hypothesis, we can take $m = K^2 n$ and $t \geq 2$. Next, we shall use N samples for the entire algorithm (without resampling), and simply apply the union bound against a failure probability of $1/n^2$, thereby giving us a high probability statement. Thus, it suffices to take $O(\max_i \mathbb{E}\left(|s_i|^5\right)^2 K^2 n \log(n)^7/\Delta^6\epsilon^2)$ samples. □

5.4.2 Experimental results

In this section, we slightly digress from our theoretical perspective and study the empirical performance of the recursive FPCA algorithm. For this purpose, we implement the algorithm as described in the previous section in Matlab. As a point of comparison, we use the FastICA algorithm [77, 75], specifically, the implementation available at [1].

The following is a full listing for the recursive partitioning algorithm for Matlab.

```
function V = recursiveFPCA(P, X, dampen)
% Fourier PCA by recursive partitioning of the space and picking best
% splits of the subspace. This is dramatically better than naive FPCA.

% This only works for n source variables and n signal variables.

% P: matrix of basis vectors
% A: Original mixing matrix we're generating samples from (we're just
%       pretending we know it).
% n: dimensionsality of the signal.
% m: number of samples

d = size(P, 2);
m = size(X,1);
```

```
if d == 1
    V = P;
    return;
end

Y = X * P;

% Now run our usual Fourier PCA routine.
u =  randn( [d,1]);
u = dampen * u / norm(u);

weights = exp( 1i * Y * u);
reweighted = bsxfun(@times, Y, weights);

zeroth = sum( weights )/m; % zeroth order
first = sum( reweighted ).'/m;
second = reweighted.' * Y/m;

psi = second * zeroth - first * first.';

[U,D] = eig(real(psi)); % U are the eigenvectors.

% Sort the eigenvalues and eigenvectors in ascending order.
[eigenvalues,perm] = sort(diag(D));
U = U(:,perm);

diff = eigenvalues(2:end) - eigenvalues(1:end-1);
[~,index] = max(diff);
```

```
smallBasis = U(:,1:index);
bigBasis = U(:,index+1:end);

smallProblem = recursiveFPCA( P * smallBasis, X, dampen);
bigProblem = recursiveFPCA( P * bigBasis, X, dampen);

V = [ smallProblem bigProblem ];
end
```

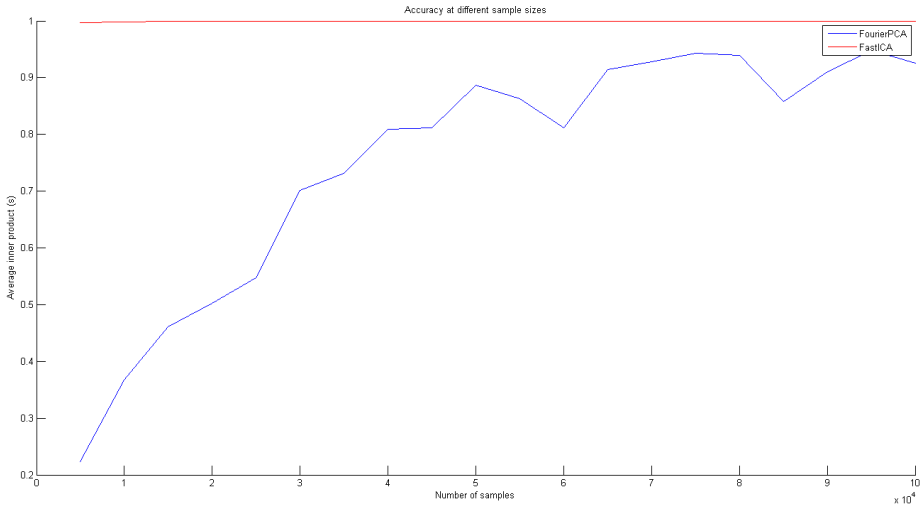
We ran two simple experiments, one with synthetic data and one with speech data (i.e., an actual blind source separation problem) using the `speech8` data from the sample benchmarks from [42]. The experiments were run on a Lenovo T420 with a dual core Intel Core i7-2620M processor clocking at 2.70 GHz, and 8 GB of RAM. The experiments were run under Matlab 7.12.0 (R2011a).

For the synthetic data, we generated the data according to a fully determined ICA model where there were $n = 100$ source variables s_i, each one of which was an independent Bernoulli $\{-1, 1\}$ random variable. The mixing matrix $A \in \mathbb{R}^{100\times 100}$ was picked as a random unitary matrix (first by generating a random matrix with Gaussian entries and then orthogonalizing its columns). Note that this system is mean zero and has unit variance, and thus one can not hope to recover A by second moment methods here.

For the speech data, we used the `speech8` dataset. We ran two experiments – in the first we simply applied a random unitary matrix as in the synthetic data. In the second, we once again apply a random unitary matrix, but this time after making each source variable isotropic. The first is meant to simulate a natural blind source separation problem, whereas the second is meant to be slightly engineered to make it impossible for second order methods to work (in fact, PCA works extremely well if we don't make the variables isotropic).

To evaluate the experiments, we constructed a bipartite matching between the true columns of A and the computed empirical columns (call them $\hat{A}$) so as to maximize the

Figure 1: Accuracy on synthetic data



average of the squared inner products:

$$\rho(A, \hat{A}) = \max_{\sigma \in S_n} \frac{1}{n} \sum_{i=1}^{n} \left\langle \hat{A}_i, A_{\sigma(i)} \right\rangle^2$$

Note that if $A = \hat{A}$, then we obtain precisely that $\rho(A, \hat{A}) = 1$. This also has desirable property that it is invariant up to permutation of columns, and sign changes in the columns. Additionally, the natural extension of this to the Hermitian inner product is in fact invariant to the complex phases of the columns.

First, for the synthetic data: in Figure 1, we can see that FastICA is a far more accurate algorithm in this ideal case, even with a few thousand samples, it is essentially perfectly accurate. On the other hand, in Figure 2, we can see that the Recursive FPCA algorithm runs faster than even FastICA (so called because of its speed). As a sidenote, we also implemented the tensor power iteration algorithm mentioned earlier in this chapter, and were unable to scale the algorithm to handle the $n = 100$ case.

For the `speech8` dataset, we had dramatically different results – in this case, the FPCA algorithm was dramatically more successful on the raw (non-isotropic) source variables, and less successful once we'd made the sources isotropic. Note that these differences can be easily detected audibly. Essentially, Fourier PCA recovery allows us to recover the source signals up the point where the author's ears are not able to distinguish between the original

Figure 2: Wall time on synthetic data

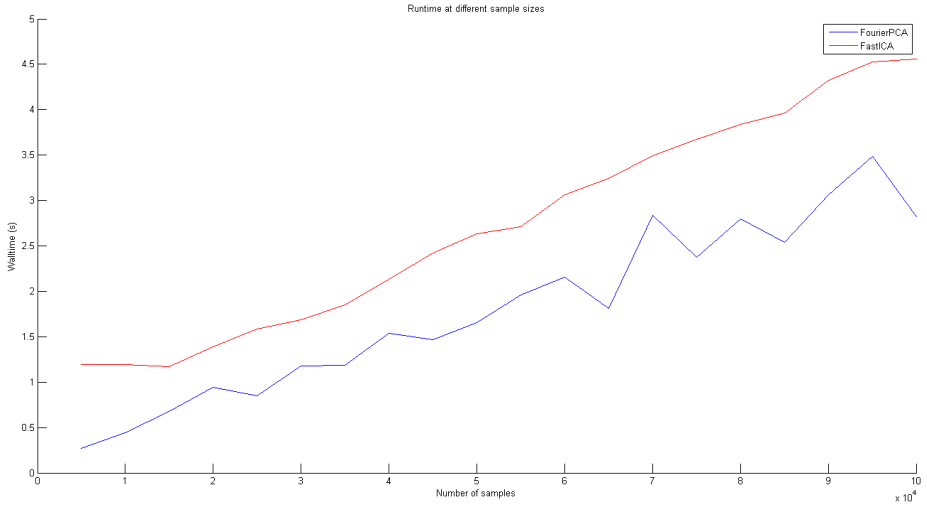


Figure 3: Accuracy on speech data

	Raw sources	Isotropic sources
Fourier PCA	0.9941	0.8324
FastICA	0.6543	0.9170

sources and recovery. Interestingly, FastICA, even when it achieves good recovery, still leads to audibly noisy/mixed recovery.

The dramatic performance of FPCA using raw (i.e., no unit variances) sources points to an interesting fact about our algorithm. Roughly speaking, most ICA algorithms employed a fixed order statistic – either second or fourth – but FPCA uses statistics of *all* orders. In particular, large differences in the variance can be very productively exploited as in this case, and it's only when the fourth cumulants are large, and second moments are unit, that we rely on the information from the fourth moment. Thus, one can view our algorithm as smoothly interpolating between second and fourth order methods in this respect.

We conclude by noting that this experimental section is by no means meant to be a comprehensive account of the practical considerations in implementing Fourier PCA, but it is meant to highlight the practical potential of the algorithm. In particular, since it is quite an efficient process to construct and diagonalize the matrix $D\psi_u$, a natural parallelization

of the algorithm is simply to split the data across multiple machines and compute approximations of A at each one using a different randomly picked u. One can then recombine the obtained columns by a simple k-means clustering algorithm, for example. These types of engineering optimisations dramatically improve the performance of the underdetermined algorithm, but their analysis lies outside of the scope of this thesis.

5.5 Mixtures of spherical Gaussians

Our work here is motivated by Hsu and Kakade's algorithm [73], which uses a tensor constructed from the first three moments of the distribution and works for a mixture of spherical Gaussians with linearly independent means.

Here we apply Fourier PCA to the classical problem of learning a mixture of Gaussians, assuming each Gaussian is spherical. More precisely, we get samples $x+\eta$, where x is from a distribution that is a mixture of k unknown Gaussians, with i'th component having mixing weight w_i and distribution $F_i = N(\mu_i, \sigma_i^2 I)$; the noise η is drawn from $N(\mu_\eta, \Sigma_\eta)$ and is not necessarily spherical. The problem is to estimate the unknown parameters w_i, μ_i, σ_i. Our method parallels the Fourier PCA approach to ICA, but here, because the structure of the problem is additive (rather than multiplicative as in ICA), we can directly use the matrix $D^2\phi$ rather than $D^2\psi = D^2 \log(\phi)$. It is easy to show that $D^2\phi = \Sigma_u$ in the description of our algorithm.

For any integrable function $f : \mathbb{C}^n \to \mathbb{C}$, we observe that for a mixture $F = \sum_{i=1}^k w_i F_i$:

$$\mathbb{E}_F\left((f(x+\eta))\right) = \sum_{i=1}^{k} w_i \mathbb{E}_{F_i}\left(f(x+\eta)\right).$$

We assume, without loss of generality, that the full mixture is centered at zero, so that:

$$\sum_{i=1}^{k} w_i \mu_i = 0$$

Fourier PCA for Mixtures

1. Pick u independently from $N(0, I_n)$.

2. Compute $M = \mathbb{E}\left(xx^T\right)$, let V be the span of its top $k-1$ eigenvectors and $\bar{\sigma}^2$ be its k'th eigenvalue and v be its k'th eigenvector. Let z be a vector orthogonal to V and to u.

3. Compute

$$\Sigma_u = \mathbb{E}\left(xx^T e^{iu^T x}\right), \quad \bar{\sigma}_u^2 = \mathbb{E}\left((z^T x)^2 e^{iu^T x}\right),$$
$$\gamma_u = \frac{1}{(u^T v)^2}\left(-\mathbb{E}\left((v^T x)^2 e^{iu^T x}\right) + \bar{\sigma}_u^2\right), \quad \tilde{u} = \mathbb{E}\left(x(z^T x)^2 e^{iu^T x}\right).$$

4. Compute the matrices

$$M = \mathbb{E}\left(xx^T\right) - \sigma^2 I \text{ and } M_u = \Sigma_u - \tilde{\sigma}_u^2 I - i\tilde{u}u^T - iu\tilde{u}^T - \gamma_u uu^T.$$

5. Run **Tensor Decomposition**(M_u, M) to obtain eigenvectors $\tilde{\mu}_j$ and eigenvalues λ_j of $M_u M^{-1}$ (in their original coordinate representation).

6. Estimate mixing weights by finding $w \geq 0$ that minimizes $\|\sum_{j=1}^k \sqrt{w_j}\tilde{\mu}_j\|$ s.t. $\sum_{j=1}^k w_j = 1$. Then estimate means and variances as

$$\mu_j = \frac{1}{\sqrt{w_j}}\tilde{\mu}_j, \qquad e^{-\frac{1}{2}\sigma_j^2\|u\|^2 + iu^T\mu_j} = \lambda_j.$$

Lemma 5.5.1. *For any $f : \mathbb{C}^n \to \mathbb{C}$, and $x \sim N(\mu, \Sigma)$ where Σ is positive definite:*

$$\mathbb{E}\left(f(x)e^{iu^T x}\right) = e^{iu^T\mu - \frac{1}{2}u^T\Sigma u}\mathbb{E}\left(f(x + i\Sigma u)\right).$$

Proof. The proof is via a standard completing the square argument; consider the exponent:

$$
\begin{aligned}
&-\frac{1}{2}\left[(x-\mu)^T\Sigma^{-1}(x-\mu)\right]+iu^Tx\\
&=-\frac{1}{2}\left[x^T\Sigma^{-1}x+\mu^T\Sigma^{-1}\mu-x^T\Sigma^{-1}\mu-\mu^T\Sigma^{-1}x\right]+iu^Tx\\
&=-\frac{1}{2}\left[x^T\Sigma^{-1}x-x^T\Sigma^{-1}(\mu+i\Sigma u)-(\mu+i\Sigma u)^T\Sigma^{-1}x+(\mu+i\Sigma u)^T\Sigma^{-1}(\mu+i\Sigma u)\right]\\
&\quad+iu^T\mu+\frac{1}{2}(\Sigma u)^T\Sigma^{-1}(\Sigma u)\\
&=-\frac{1}{2}(x-(\mu+i\Sigma u))^T\Sigma^{-1}(x-(\mu+i\Sigma u))+iu^T\mu-\frac{1}{2}u^T\Sigma u
\end{aligned}
$$

Thus:

$$
\begin{aligned}
&\mathbb{E}\left(f(x)e^{iu^Tx}\right)\\
&=\frac{1}{\det(\Sigma)^{1/2}(2\pi)^{n/2}}\int f(x)e^{iu^Tx}e^{-\frac{1}{2}(x-\mu)^T\Sigma^{-1}(x-\mu)}dx\\
&=\frac{1}{\det(\Sigma)^{1/2}(2\pi)^{n/2}}\int f(x)e^{-\frac{1}{2}(x-(\mu+i\Sigma u))^T\Sigma^{-1}(x-(\mu+i\Sigma u))}e^{iu^T\mu-\frac{1}{2}u^T\Sigma u}dx
\end{aligned}
$$

Now with a change of variables $y=x-i\Sigma u$, we obtain:

$$
\begin{aligned}
\mathbb{E}\left(f(x)e^{iu^Tx}\right)&=\frac{1}{\det(\Sigma)^{1/2}(2\pi)^{n/2}}e^{iu^T\mu-\frac{1}{2}u^T\Sigma u}\int f(y+i\Sigma u))e^{-\frac{1}{2}(y-\mu)^T\Sigma^{-1}(y-\mu)}dy\\
&=e^{iu^T\mu-\frac{1}{2}u^T\Sigma u}\mathbb{E}\left(f(y+i\Sigma u)\right)
\end{aligned}
$$

□

Note: technically we require that $\mathbb{E}(|f(x)|)<\infty$ with respect to a Gaussian measure so as to apply the dominated convergence theorem, and an analytic extension of the Gaussian integral to complex numbers, but these arguments are standard and we omit them (see for example [92]).

Lemma 5.5.2. *Let $x\in\mathbb{R}^n$ be drawn from a mixture of k spherical Gaussians in $\mathbb{R}^n$, $u,z\in\mathbb{R}^n$ as in the algorithm. Let $\hat{w}_j=w_je^{iu^T\mu_j-\frac{1}{2}\sigma_j^2\|u\|^2}$. Then,*

$$\mathbb{E}\left(xx^T\right)=\sum_{j=1}^k w_j\sigma_i^2I+\sum_{j=1}^k w_j\mu_j\mu_j^T. \tag{28}$$

$$\mathbb{E}\left(xx^Te^{iu^Tx}\right)=\sum_{j=1}^k \hat{w}_j\sigma_j^2I+\sum_{j=1}^k \hat{w}_j(\mu_j+i\sigma_j^2u)(\mu_j+i\sigma_j^2u)^T. \tag{29}$$

$$\mathbb{E}\left(x(z^Tx)^2e^{iu^Tx}\right)=\sum_{j=1}^k \hat{w}_j\sigma_j^2(\mu_i+i\sigma_j^2u)\ . \tag{30}$$

Proof. These are obtained by direct calculation and expanding out $x_i \sim N(\mu_i, \sigma^2 I_n)$. For (28):

$$\begin{aligned}\mathbb{E}\left(xx^T\right) &= \sum_{j=1}^{k} w_j \mathbb{E}_{F_j}\left(xx^T\right) \\ &= \sum_{j=1}^{k} w_j \mathbb{E}\left((x-\mu_j+\mu_j)(x-\mu_j+\mu_j)^T\right) \\ &= \sum_{j=1}^{k} w_j \left[\sigma_j^2 I_n + \mu_j \mu_j^T\right]\end{aligned}$$

(29) follows by applying Lemma 5.5.1 and the previous result:

$$\begin{aligned}\mathbb{E}\left(xx^T e^{iu^T x}\right) &= \sum_{i=1}^{k} w_i e^{iu^T\mu_i - \frac{1}{2}\sigma_i^2\|u\|^2} \mathbb{E}_{F_i}\left((x_i+i\sigma_i^2 u)(x_i+i\sigma_i^2 u)^T\right) \\ &= \sum_{i=1}^{k} \hat{w}_i \left[\sigma_i^2 I_n + (\mu_i+i\sigma_i^2 u)(\mu_i+i\sigma_i^2 u)^T\right]\end{aligned}$$

To see (30), we write (noting that z is orthogonal to u and to each μ_j),

$$\begin{aligned}\mathbb{E}\left(x(z^T x)^2 e^{iu^T x}\right) &= \sum_{j=1}^{k} \hat{w}_j \mathbb{E}\left((x+i\sigma_j^2 u)(z^T(x+i\sigma_j^2 u))^2\right) \\ &= \sum_{j=1}^{k} \hat{w}_j \mathbb{E}\left((x-\mu_j+\mu_j+i\sigma_j^2 u)(z^T(x-\mu_j))^2\right) \\ &= \sum_{j=1}^{k} \hat{w}_j \left(\mathbb{E}\left((x-\mu_j)(z^T(x-\mu_j))^2\right) + \mathbb{E}\left((\mu_j+i\sigma_j^2 u)(z^T(x-\mu_j))^2\right)\right) \\ &= \sum_{j=1}^{k} \hat{w}_j \sigma_j^2 (\mu_j + i\sigma_j^2 u).\end{aligned}$$

□

Instead of polynomial anti-concentration under a Gaussian measure, we require only a simpler lemma concerning the anti-concentration of complex exponentials:

Lemma 5.5.3 (Complex exponential anti-concentration)**.** *Let $\mu_i, \mu_j \in \mathbb{R}^n$ satisfy $\|\mu_i - \mu_j\| > 0$, then for $u \sim N(0, \sigma^2 I_n)$ where $\|\mu_i - \mu_j\|^2 \sigma^2 \leq 2\pi^2$. Then:*

$$\mathsf{Pr}\left(\left|e^{i\mu_i^T u} - e^{i\mu_j^T u}\right| \leq \epsilon\right) \leq \frac{16\epsilon}{\|\mu_i - \mu_j\| \sigma\sqrt{2\pi}}.$$

Proof. First, note that it suffices to show anti-concentration of the complex exponential around 1:

$$\left|e^{i\mu_i^T u} - e^{i\mu_j^T u}\right| = \left|e^{i\mu_i^T u}\right| \left|1 - e^{i(\mu_i-\mu_j)^T u}\right| = \left|1 - e^{i(\mu_i-\mu_j)^T u}\right|$$

The exponent $(\mu_i - \mu_j)^T u$ is of course a random variable $z \in \mathbb{R}$ distributed according to $N(0, \sigma^2 \left\|\mu_i - \mu_j\right\|^2)$. From plane geometry, we know that: $\left|e^{iz} - 1\right| > \epsilon$ in case

$$z \notin \cup_{k\in\mathbb{Z}}[2\pi k - 2\epsilon, 2\pi k + 2\epsilon]$$

We can bound this probability as follows:

$$\begin{aligned}
\mathsf{Pr}\left(z \notin \cup_{k\in\mathbb{Z}}[2\pi k - 2\epsilon, 2\pi k + 2\epsilon]\right) &\leq 2\sum_{k=0}^{\infty} \frac{4\epsilon}{\left\|\mu_i - \mu_j\right\| \sigma\sqrt{2\pi}} e^{-\frac{(2\pi k)^2}{2\left\|\mu_i-\mu_j\right\|^2\sigma^2}} \\
&\leq \frac{8\epsilon}{\left\|\mu_i - \mu_j\right\| \sigma\sqrt{2\pi}} \sum_{k=0}^{\infty} e^{-\frac{2\pi^2 k}{\left\|\mu_i-\mu_j\right\|^2\sigma^2}} \\
&= \frac{8\epsilon}{\left\|\mu_i - \mu_j\right\| \sigma\sqrt{2\pi}} \frac{1}{1 - e^{-\frac{2\pi^2}{\left\|\mu_i-\mu_j\right\|^2\sigma^2}}} \\
&\leq \frac{16\epsilon}{\left\|\mu_i - \mu_j\right\| \sigma\sqrt{2\pi}}
\end{aligned}$$

where the last line follows from the assumption $\left\|\mu_i - \mu_j\right\|^2 \sigma^2 \leq 2\pi^2$. □

We can now prove that the algorithm is correct with sufficiently many samples. Using PCA we can find the span of the means $\{\mu_1, \ldots, \mu_k\}$, as the span of the top $k-1$ right singular vectors of the matrix whose rows are sample points [127]. Projecting to this space, the mixture remains a mixture of spherical Gaussians. We assume that the μ_i are linearly independent (as in recent work [73] with higher moments).

Proof of Theorem 5.1.1. From Lemma 5.5.2, we observe that for any unit vector v,

$$\mathbb{E}\left((v^T x)^2\right) = v^T \mathbb{E}\left(xx^T\right) v = \sum_{i=1}^{k} w_i \sigma_i^2 + \sum_{i=1}^{k} w_i (\mu_i^T v)^2.$$

Without loss of generality, we may assume that the overall mean is 0, hence $0 = \sum_i w_i \mu_i$ is 0 and therefore the μ_i are linearly dependent, and there exist a v orthogonal to all the μ_i. For such a v, the variance is $\sigma^2 = \sum_{i=1}^{k} w_i \sigma_i^2$ while for v in the span, the variance is

strictly higher. Therefore the value σ^2 is simply the k'th eigenvalue of $\mathbb{E}\left(xx^T\right)$ (assuming x is centered at 0).

Thus, in the algorithm we have estimated the matrices

$$M = \sum_{i=1}^{k} w_i \mu_i \mu_i^T = AA^T \text{ and } M_u = \sum_{i=1}^{k} w_i e^{-\frac{1}{2}\|u\|^2\sigma_i^2 + iu^T\mu_i} \mu_i \mu_i^T = AD_uA^T.$$

with $(D_u)_{ii} = e^{-\frac{1}{2}\|u\|^2\sigma_i^2 + iu^T\mu_i}$. Thus,

$$M_u M^{-1} = AD_uA^{-1}$$

and its eigenvectors are the columns of A, assuming the entries of D_u are distinct. We note that the columns of A are precisely $\tilde{\mu}_j = \sqrt{w_j}\mu_j$. The eigenvalue corresponding to the eigenvector $\tilde{\mu}_j$ is the j'th diagonal entry of D_u.

To prove the algorithm's correctness, we will once again apply Theorem 4.3.5 for robust tensor decomposition by verifying its five conditions. Condition 1 holds by our assumption on the means of the Gaussian mixtures. Conditions 3 holds by taking sufficiently many samples (the overall sample and time complexity will be linear in n and polynomial in k), Conditions 2 and 4 hold by applying 5.5.3. □

We can apply our observations regarding Gaussian noise from Section 5.3.5. Namely, the covariance of the reweighted Gaussian is shifted by Σ_η, the covariance of the unknown noise. Thus, by considering Σ_u and the standard covariance, and taking their difference, the contribution of the noise is removed and we are left with a matrix that can be diagonalized using A.

5.6 Underdetermined ICA

5.6.1 Overview

In this section we give our algorithm for the underdetermined ICA problem and analyze it. Additional assumptions are needed for the essentially unique identifiability of this model. For example, suppose that columns A_i and A_j are parallel i.e., $A_i = cA_j$, then one could replace the variables s_i and s_j with $s_i + cs_j$ and 0 and the model would still be consistent. We introduce the following sufficient condition for identifiability: we require that the m

column vectors of $A^{\odot k}$ be linearly independent for some $k > 0$ (smaller k would be better for the efficiency of the algorithm). We make this quantitative by requiring that the m'th singular value satisfy $\sigma_m(A^{\odot k}) > 0$.

Our approach to the underdetermined ICA problem is to apply our tensor decomposition to a pair of carefully-chosen tensors that arise from the distribution. The tensors we use are the derivative tensors of the second charateristic function $\psi_x(u) = \log\left(\mathbb{E}\left(e^{iu^T x}\right)\right)$.

This method generalises the fourth moment methods for ICA where one computes the local optima of the following quartic form:

$$f(u) = \mathbb{E}\left((x^T u)^4\right) - 3\mathbb{E}\left((x^T u)^2\right)^2.$$

An equivalent formulation of this is to consider the tensor $T \in \mathbb{R}^{n\times n\times n\times n}$ which represents this quartic form (just as in the matrix case where symmetric matrices represent quadratic forms, symmetric tensors of order 4 represent quartic forms). Let us denote our overall tensor representing $f(u)$ by T where $f(u) = T(u, u, u, u)$. By a relatively straightforward calculation, one can verify that $T(u, u, u, u)$ is the fourth derivative of the second characteristic function of x evaluated at 0:

$$T = D_u^4 \psi_x(0).$$

On the other hand, one can also verify that T has the following decomposition (see for example [9]):

$$T = \sum_{j=1}^{m} \left(\mathbb{E}\left(s_i^4\right) - 3\mathbb{E}\left(s_i^2\right)^2\right) A_i \otimes A_i \otimes A_i \otimes A_i$$

So in fact, one can view the fourth moment tensor methods as performing the tensor decomposition of only one tensor – the fourth derivative of ψ evaluated at 0!

Our method also generalises the algorithm we gave for the fully determined case in Section 5.3. We can view that case as simply being the second derivative version of the more general algorithm. The techniques used in this section are generalisations and refinements of those used in the fully determined case, though replacing the easy matrix decomposition arguments with the corresponding (harder) tensor arguments.

A property of the second characteristic function that is central for our algorithm is that the second characteristic function of a random vector with independent components factorizes into the sum of the second characteristic functions of each component:

$$\log\left(\mathbb{E}\left(e^{iu^T s}\right)\right) = \sum_{j=1}^{m} \log\left(\mathbb{E}\left(e^{iu_j s_j}\right)\right),$$

and now every mixed partial derivative (with respect to u_j and $u_{j'}$) is 0, as each term in the sum depends only on one component of u. Taking the derivative tensor will result in a diagonal tensor where the offdiagonal terms are all 0. In the case when we're interested in $x = As$, we simply need to perform the change of basis via A very carefully for the derivative tensors via the chain rule. One could also try to perform this analysis with the moment generating function $\mathbb{E}\left(e^{u^T x}\right)$ without the complex phase. The difficulty here is that the moment generating functions exists only if all moments of x exist, and thus a moment generating function approach would not be able to deal with heavy tailed distributions. Moreover, using a real exponential leads us to estimate exponentially large quantities from samples, and it is difficult to get good bounds on the sample complexity. Using the complex exponential avoids these problems as all quantities have modulus 1.

We will then apply our tensor decomposition framework: as before we show that the eigenvalues of the flattened derivative tensors are well spaced in Section 5.6.4. We then study the sample complexity in Section 5.6.5 and subsequently assemble these components into the main theorem.

5.6.2 Fourier derivatives

We begin by analysing the derivative tensor $D^r\psi_u$ and its flattening to a matrix. At order r, $D^r\psi_u$ will be an order r tensor in $\mathbb{R}^{n\times\cdots\times n}$ evaluated at the point u. In particular, we will take the view that this derivative tensor defines an r-homogenous form over vectors $v \in \mathbb{R}^n$ i.e.,

$$(D^r f_u)(v,\ldots,v) = \sum_{i_1,\ldots,i_r} (D^r f_u)_{i_1,\ldots,i_r} v_{i_1}\cdots v_{i_r}$$

and in particular when $r = 1$, the derivative Df_u lies in the dual space of $\mathbb{R}^n$ and can be represented by a row vector. What we prove in this section essentially amounts to the chain

rule for higher derivatives. As an example, we start with the first derivative:

Lemma 5.6.1. *Let $f : \mathbb{R}^n \to \mathbb{R}$ be a differentiable function and let $A \in \mathbb{R}^{n\times m}$ be a linear map. Then for the composite function $f(Au)$:*

$$D(f(Au)) = (Df_{Au})A$$

Note that Df_{Au} lies in the dual space of $\mathbb{R}^n$ and is a row vector whose i^{th} coordinate is given by:

$$(Df_{Au})_i = \left.\frac{\partial f(x)}{\partial x_i}\right|_{x=Au}$$

The proof is trivial and is a direct consequence of the chain rule. We will also use this to compute higher derivatives.

Lemma 5.6.2. *Let $x = As$ be drawn from an ICA model, then for $d \in 2\mathbb{N}$:*

$$M_u = \left[\text{vec}\left(A_i^{\otimes d/2}\right)\right] \text{diag}\left(\psi_i^{(d)}(A_i^T u)\right) \left[\text{vec}\left(A_i^{\otimes d/2}\right)\right]^T$$

Proof. Let $H_y^r \in \mathbb{C}^{m\times\cdots\times m}$ denote the r^{th} derivative tensor of $\zeta = \mathbb{E}\left(e^{it^T s}\right)$ evaluated at the point $y \in \mathbb{R}^m$; note that this characteristic function is taken with respect to the random vector s and not x, and that $\psi(u) = \zeta(A^T u)$. Note that if we allow u to vary then we obtain a tensor field – a map from $\mathbb{R}^n$ to $\mathbb{R}^{m\times\cdots\times m}$ defined by H^r evaluated at $A^T u$ (i.e. when $r = 1$, the gradient defines a vector field over $\mathbb{R}^n$). As an abuse of notation, we will denote this tensor field by $H^r_{A^T u}$ as a function of u.

We will show for all $r \in \mathbb{N}$ that:

$$D^r\psi_u = D^r(\zeta \circ A^T)_u = H^r_{A^T u}(A^T\cdot, \ldots, A^T\cdot) \tag{31}$$

To obtain Equation 31, we can induct over r. Consider the base case where $r = 1$, by Lemma 5.6.1 we have that:

$$D\psi_u = D\zeta(A^T u) = D\zeta_{A^T u}A^T = H^1_{A^T u}A^T = H^1_{A^T u}(A^T\cdot)$$

Assume the inductive hypothesis. Next, we want to compute tensor $D^{r+1}\psi_u$. At entry $(i_1, \ldots, i_r, j)$ this is the the partial derivative with respect to u_j of the $(i_1, \ldots, i_r)$ entry of $D^r\psi_u$. Thus, we want to compute:

$$\frac{\partial}{\partial u_j}[H^r_{A^T u}(A^T\cdot, \ldots, A^T\cdot)]_{i_1,\ldots,i_r} = \frac{\partial}{\partial u_j}\left(\sum_{i'_1,\ldots,i'_r}(H^r_{A^T u})_{i'_1,\ldots,i'_r}A_{i_1,i'_1}\cdots A_{i_r,i'_r}\right)$$

If we examine one term of this sum by passing the partial derivative operator inside the sum, then we observe that this is simply the j'th coordinate of the derivative of a function which takes $\mathbb{R}^m$ as its domain and $\mathbb{R}$ as its image. Thus, we can apply Lemma 5.6.1 again:

$$\begin{aligned}\frac{\partial}{\partial u_j}(H^r_{A^T u})_{i'_1,\ldots,i'_r} &= [D(H^r_{A^T u})_{i'_1,\ldots,i'_r}]_j \\ &= \left[(H^{r+1}_{A^T u})_{i'_1,\ldots,i'_r}A^T\right]_j \\ &= \sum_{j'}(H^{r+1}_{A^T u})_{i'_1,\ldots,i'_r,j'}A_{j,j'}\end{aligned}$$

Thus we have for all coordinates:

$$[H^r(A^T\cdot, \ldots, A^T\cdot)]_{i_1,\ldots,i_r,j} = \sum_{i'_1,\ldots,i'_r,j'}(H^j_{A^T u})_{i'_1,\ldots,i'_r,j'}A_{i_1,i'_1}\cdots A_{i_r,i'_r}A_{j,j'}$$

One can then trivially verify entry-wise that for a diagonal tensor H^r that the flattening relationship holds. □

5.6.3 Algorithm

For underdetermined ICA we compute the higher derivative tensors of the second characteristic function $\psi_x(u) = \log(\phi_x(u))$ at two random points and run the tensor decomposition algorithm from the previous section.

Underdetermined ICA(σ)

1. (Compute derivative tensor) Pick independent random vectors $\alpha, \beta \sim N(0, \sigma^2 I_n)$. For even d, estimate the d^{th} derivative tensors of $\psi_x(u)$ at α and β as $T_\alpha = D^d_u\psi_x(\alpha)$ and $T_\beta = D^d_u\psi_x(\beta)$.

2. (Tensor decomposition) Run **Tensor Decomposition**(T_α, T_β).

To estimate the $2d^{th}$ derivative tensor of $\psi_x(u)$ empirically, one simply writes down the expression for the derivative tensor, and then estimates each entry from samples using the naive estimator.

More precisely, we can use

$$\frac{\partial \phi(u)}{\partial u_i} = \mathbb{E}\left((ix_i)e^{iu^T x}\right).$$

This states that differentiation in the Fourier space is equivalent to multiplication in the original space, thus it suffices to estimate monomials of x reweighted by complex exponentials. To estimate the d^{th} derivative tensor of $\log(\phi(u))$ empirically, one simply writes down the expression for the derivative tensor, and then estimates each entry from samples using the naive estimator. Note that the derivatives can be somewhat complicated, for example,

at fourth order we have

$$
\begin{aligned}
&[D^4\psi_u]_{i_1,i_2,i_3,i_4} \\
&= \frac{1}{\phi(u)^4}\left[\mathbb{E}\left((ix_{i_1})(ix_{i_2})(ix_{i_3})(ix_{i_4})\exp(iu^Tx)\right)\phi(u)^3\right. \\
&\quad - \mathbb{E}\left((ix_{i_2})(ix_{i_3})(ix_{i_4})\exp(iu^Tx)\right)\mathbb{E}\left((ix_{i_1})\exp(iu^Tx)\right)\phi(u)^2 \\
&\quad - \mathbb{E}\left((ix_{i_2})(ix_{i_3})\exp(iu^Tx)\right)\mathbb{E}\left((ix_{i_1})(ix_{i_4})\exp(iu^Tx)\right)\phi(u)^2 \\
&\quad - \mathbb{E}\left((ix_{i_2})(ix_{i_4})\exp(iu^Tx)\right)\mathbb{E}\left((ix_{i_1})(ix_{i_3})\exp(iu^Tx)\right)\phi(u)^2 \\
&\quad - \mathbb{E}\left((ix_{i_2})\exp(iu^Tx)\right)\mathbb{E}\left((ix_{i_1})(ix_{i_3})(ix_{i_4})\exp(iu^Tx)\right)\phi(u)^2 \\
&\quad - \mathbb{E}\left((ix_{i_3})(ix_{i_4})\exp(iu^Tx)\right)\mathbb{E}\left((ix_{i_1})(ix_{i_2})\exp(iu^Tx)\right)\phi(u)^2 \\
&\quad - \mathbb{E}\left((ix_{i_3})\exp(iu^Tx)\right)\mathbb{E}\left((ix_{i_1})(ix_{i_2})(ix_{i_4})\exp(iu^Tx)\right)\phi(u)^2 \\
&\quad - \mathbb{E}\left((ix_{i_4})\exp(iu^Tx)\right)\mathbb{E}\left((ix_{i_1})(ix_{i_2})(ix_{i_3})\exp(iu^Tx)\right)\phi(u)^2 \\
&\quad + 2\mathbb{E}\left((ix_{i_3})(ix_{i_4})\exp(iu^Tx)\right)\mathbb{E}\left((ix_{i_2})\exp(iu^Tx)\right)\mathbb{E}\left((ix_{i_1})\exp(iu^Tx)\right)\phi(u) \\
&\quad + 2\mathbb{E}\left((ix_{i_3})\exp(iu^Tx)\right)\mathbb{E}\left((ix_{i_2})(ix_{i_4})\exp(iu^Tx)\right)\mathbb{E}\left((ix_{i_1})\exp(iu^Tx)\right)\phi(u) \\
&\quad + 2\mathbb{E}\left((ix_{i_4})\exp(iu^Tx)\right)\mathbb{E}\left((ix_{i_2})(ix_{i_3})\exp(iu^Tx)\right)\mathbb{E}\left((ix_{i_1})\exp(iu^Tx)\right)\phi(u) \\
&\quad + 2\mathbb{E}\left((ix_{i_3})\exp(iu^Tx)\right)\mathbb{E}\left((ix_{i_2})\exp(iu^Tx)\right)\mathbb{E}\left((ix_{i_1})(ix_{i_4})\exp(iu^Tx)\right)\phi(u) \\
&\quad + 2\mathbb{E}\left((ix_{i_4})\exp(iu^Tx)\right)\mathbb{E}\left((ix_{i_2})\exp(iu^Tx)\right)\mathbb{E}\left((ix_{i_1})(ix_{i_3})\exp(iu^Tx)\right)\phi(u) \\
&\quad + 2\mathbb{E}\left((ix_{i_4})\exp(iu^Tx)\right)\mathbb{E}\left((ix_{i_3})\exp(iu^Tx)\right)\mathbb{E}\left((ix_{i_1})(ix_{i_2})\exp(iu^Tx)\right)\phi(u) \\
&\quad \left. -6\mathbb{E}\left((ix_{i_1})\exp(iu^Tx)\right)\mathbb{E}\left((ix_{i_2})\exp(iu^Tx)\right)\mathbb{E}\left((ix_{i_3})\exp(iu^Tx)\right)\mathbb{E}\left((ix_{i_4})\exp(iu^Tx)\right)\right].
\end{aligned}
$$

The salient points are described in Lemma 5.3.6 and Claim 5.3.7: there are at most $2^{d-1}(d-1)!$ terms (counting multiplicities), and no term has combined exponents of x_i in all it factors higher than d. We will give a rigorous analysis of the sampling error incurred in Section 5.6.5.

The analysis roughly proceeds as follows: by Lemma 5.6.2 for tensors flattened into matrices we have $D_u^{2d}\psi_x(\alpha) = A^{\odot d}\mathrm{diag}\left(\partial_{t_1}^{2d}\psi_s(A^T\alpha),\ldots,\partial_{t_1}^{2d}\psi_s(A^T\alpha)\right)(A^{\odot d})^T$ and $D_u^{2d}\psi_x(\beta) = A^{\odot d}\mathrm{diag}\left(\partial_{t_1}^{2d}\psi_s(A^T\beta),\ldots,\partial_{t_1}^{2d}\psi_s(A^T\beta)\right)(A^{\odot d})^T$.

Thus we have two tensors with shared rank-1 factors as in the tensor decomposition algorithm above. For our tensor decomposition to work, we require that all the ratios

$(\partial_{t_j}^{2d}\psi_s(A^T\alpha))/(\partial_{t_j}^{2d}\psi_s(A^T\beta))$ for $j \in [m]$ be different from each other as otherwise the eigenspaces in the flattened forms will mix and we will not be able to uniquely recover the columns A_i. To this end, we will express $\partial_{t_j}^{2d}\psi_s(A^T\alpha)$ as a low degree polynomial plus error term (which we will control by bounding the derivatives of ψ_s). The low degree polynomials will with high probability take on sufficiently different values for A^Tu and A^Tv, which in turn guarantees that their ratios, even with the error terms, are quite different.

Our analysis for both parts might be of interest for other problems. On the way to doing this in full generality for underdetermined ICA, we first consider the special case of $d = 2$, which will already involve several of these concepts and the algorithm itself is just PCA reweighted with Fourier weights.

5.6.4 Eigenvalue spacings

In this subsection we examine the anti-concentration of the diagonal entries $\psi_i^{(d)}((A^Tu)_i)$. The analysis has similarities to the fully-determined case but there are also some major differences: in the fully-determined case, A_i^Tu and A_j^Tu are independent Gaussians because the columns of A are orthogonal by isotropic position (recall that we defined A_i^T to mean $(A_i)^T$). We can not make A an orthonormal matrix in the underdetermined case, so we have to exploit the more limited randomness. First, let us pursue the same strategy as in the fully determined case and first control the truncationerror.

Lemma 5.6.3. *Let $s = (s_1, \ldots, s_m) \in \mathbb{R}^m$ be a random vector with indpendent components each with finite k absolute moments, and for $t \in \mathbb{R}^m$ let $\phi(t) = \mathbb{E}\left(e^{it^Ts}\right)$ be the associated characteristic function. Then for $k \geq 1$ and $i_1, \ldots i_k \in [m]$ we have*

$$|\partial_{i_1,\ldots,i_k} \log \phi(t)| \leq \frac{2^{k-1}(k-1)! \max_{j\in[m]} \mathbb{E}\left(|s_j|^k\right)}{|\phi(t)|^k}.$$

Proof. To compute the derivatives of $\log \phi(t)$ we proceed inductively with $\partial_{i_1} \log \phi(t) = (\partial_{i_1}\phi(t))/\phi(t)$ as our base case. For $d < k$, write $\partial_{i_1,\ldots,i_d}(\log \phi)$ as $N_d(t)/\phi(t)^d$. Then we have

$$\begin{aligned}\partial_{i_1,\ldots,i_{d+1}} \log \phi(t) &= \partial_{i_{d+1}} \left(\frac{N_d(t)}{\phi(t)^d} \right) \\ &= \frac{(\partial_{i_{d+1}} N_d(t))\phi(t)^d - N_d(t) d \phi(t)^{d-1} \partial_{i_{d+1}} \phi(t)}{\phi(t)^{2d}} \\ &= \frac{(\partial_{i_{d+1}} N_d(t))\phi(t) - d N_d(t) \partial_{i_{d+1}} \phi(t)}{\phi(t)^{d+1}}.\end{aligned} \tag{32}$$

We make the following claim about $N_d(t)$:

Claim 5.6.4. *The functions $N_d(t)$ is the sum of terms of the form $C_{S_1,\ldots,S_d} \partial_{S_1} \ldots \partial_{S_d} \phi(t)$ where multisets $S_1, \ldots, S_d \subseteq \{i_1, \ldots, i_d\}$ (this is a multiset) satisfy $S_1 \cup \ldots \cup S_d = \{i_1, \ldots, i_d\}$, and $C_{S_1,\ldots,S_d}$ are integer coefficients with $\sum |C_{S_1,\ldots,S_d}| \leq 2^{d-1}(d-1)!$.*

Proof. The first part follows via induction on d and (32). For the second part, let $T(d)$ denote $\sum |C_{S_1,\ldots,S_d}|$. Note that $T(1) = 1$. Then by (32), we have $T(d+1) \leq dT(d) + dT(d)$, which gives $T(d) \leq 2^{d-1}(d-1)!$. □

Returning to the proof of Lemma 5.6.3, we observe that for any multiset S with elements from $[m]$ and size at most k, we have

$$|\partial_S \phi(t)| = \left| \mathbb{E}\left(i^{|S|} s_S e^{it^T s} \right) \right| \leq \mathbb{E}\left(|s_S| \right).$$

For $\ell \in [m]$, let p_ℓ be the number of times ℓ occurs in the multiset $\{i_1, \ldots, i_d\}$. For each term of $N_d(t)$ we have

$$\begin{aligned}\left| \prod_{j=1}^{d} \partial_{S_j} \phi \right| &= \prod_{j=1}^{d} \left| \partial_{S_j} \phi \right| \\ &\leq \prod_{j=1}^{d} \mathbb{E}\left(\left| s_{S_j} \right| \right) \\ &= \prod_{\ell=1}^{m} \mathbb{E}\left(|s_\ell|^{p_\ell} \right) \\ &\leq \prod_{\ell=1}^{m} \left(\mathbb{E}\left(|s_\ell|^d \right) \right)^{p_\ell/d} \\ &\leq \max_{\ell \in [m]} \mathbb{E}\left(|s_\ell|^d \right).\end{aligned} \tag{33}$$

The second equality above uses the independence of the s_ℓ, and the second inequality uses the first part of Fact 3.1.1.

Thus $|N_d(t)| \leq 2^{(d-1)}(d-1)! \max_{\ell \in [m]} \mathbb{E}\left(|s_\ell|^d\right)$, which when divided by $\phi(t)^d$ gives the required bound. $\square$

An additional complication in the underdetermined case is that we are working with anti-concentration of the quotients of such diagonal entries rather than the entries themselves.

Theorem 5.6.5. *Let $s \in \mathbb{R}^m$ be a random vector with independent components. For $t \in \mathbb{R}^m$ and $d \in 2\mathbb{N}$ let $\psi_a(t) := \log \mathbb{E}\left(e^{it_a s_a}\right)$, and $g_a(t_a) := \frac{d^d \psi_a(t_a)}{dt_a^d}$ for all $a \in [m]$. Let $0 < \delta$. Suppose that the following data and conditions are given:*

1. *$\mathbb{E}(s_a) = 0$, $\mathbb{E}\left(s_a^2\right) \leq M_2$ and $\mathbb{E}\left(s_a^d\right) \leq M_d$ for $a \in [m]$ and $M_2 < M_d$.*
2. *$k \geq 2$ and for all $a \in [m]$, $k_a \in \mathbb{N}$ where $d < k_a < k$, such that $|\mathrm{cum}_{k_a}(s_a)| \geq \Delta$.*
3. *$\mathbb{E}\left(|s_a|^{k_a+1}\right) \leq M_k$ for $a \in [m]$ and $M_2 < M_k$.*
4. *$A \in \mathbb{R}^{n \times m}$ be a full row rank matrix whose columns all have unit norm and $1 - \langle A_a, A_b \rangle^2 \geq L^2$ for all pairs of columns.*
5. *$u, v \sim N(0, \sigma^2 I_n)$ sampled independently where*

$$\sigma \leq \min\left(1, \frac{1}{2\sqrt{2M_2 \log 1/q}}, \sigma'\right),$$

and

$$\sigma' = \Delta \frac{k-d+1}{k!} \left(\frac{3}{8}\right)^k \frac{1}{M_k} \left(\frac{Lq\sqrt{2\pi}}{4(k-d)}\right)^{k-d} \left(\frac{1}{\sqrt{2\log 1/q}}\right)^{k-d}$$

and $0 < q < 1/3$. Then with probability at least $1 - 3\binom{m}{2}q$ we have

$$\left|\frac{g_b(A_b^T u)}{g_b(A_b^T v)} - \frac{g_a(A_a^T u)}{g_a(A_a^T v)}\right| \geq \Delta \frac{1}{(k-d)!(d-1)!} \left(\frac{3}{8}\right)^d \frac{1}{M_d} \left(\frac{\sigma L q \sqrt{2\pi}}{4(k-d)}\right)^{(k-d)}, \tag{34}$$

for all distinct $a, b \in [m]$. (We count the small probability case where $g_b(A_b^T v) = 0$ or $g_a(A_a^T v) = 0$ as violating the event in (34).)

Proof. Fix $a \neq b \in [m]$ and show that the spacing in (34) is lower bounded for this pair with high probability. We will then take a union bound over all $\binom{m}{2}$ pairs, which will give the desired result.

For random choice of v, the events

$$g_a(A_a^T v) \neq 0 \text{ and } g_b(A_b^T v) \neq 0 \tag{35}$$

have probability 1. Thus in the following we will assume that these events are true.

We will need concentration of $(A_a^T u)$ and of $(A_a^T v)$.

$$\mathsf{Pr}_{u\sim N(0,\sigma^2)}\left(\left|A_a^T u\right| > \tau\right) \leq \sqrt{\frac{2}{\pi}}\sigma^2 \left\|r\right\|^2 \frac{1}{\tau} e^{-\frac{\tau^2}{2\sigma^2\|r\|^2}} \leq \sqrt{\frac{2}{\pi}}\sigma^2 \frac{1}{\tau} e^{-\frac{\tau^2}{2\sigma^2}},$$

here the first inequality used Claim 3.1.2 and the second used the fact that $\|r\| \leq 1$. Substituting $\tau = \sigma\sqrt{2\log 1/q}$ we get

$$\mathsf{Pr}\left(\left|A_a^T u\right| \leq \sigma\sqrt{2\log 1/q}\right) \geq 1 - \frac{\sigma q}{\sqrt{\pi \log 1/q}} \geq 1 - \frac{q}{\sqrt{\pi \log 1/q}}.$$

By the union bound we have

$$\mathsf{Pr}\left(\left|A_a^T u\right|, \left|A_a^T v\right| \leq \sigma\sqrt{2\log 1/q}\right) \geq 1 - \frac{2q}{\sqrt{\pi \log 1/q}}. \tag{36}$$

In the sequel we will assume that the event in the previous expression happens.

Under the assumption that $\left|A_a^T v\right| \leq \sigma\sqrt{2\log 1/q}$ we have

$$\left|g_a(A_a^T v)\right| = \left|\psi^{(d)}(A_a^T v)\right| \leq \frac{2^{d-1}(d-1)! M_d}{(3/4)^d}, \tag{37}$$

where to upper bound $\left|\psi^{(d)}(A_a^T u)\right|$ we used Lemma 5.3.6, Lemma 5.3.4, and the condition $\sigma\sqrt{2\log 1/q} \leq \frac{1}{2\sqrt{M_2}}$ which follows from our assumption on σ.

To compute the probability that the spacing is at least ϵ_a, where $\epsilon_a > 0$ will be chosen later, we condition on fixing of $A_b^T u = z$ and any fixing of v:

$$\mathsf{Pr}\left(\left|\frac{g_a(A_a^T u)}{g_a(A_a^T v)} - \frac{g_b(A_b^T u)}{g_b(A_b^T v)}\right| \leq \epsilon_a\right) = \int \mathsf{Pr}\left(\left|\frac{g_a(A_a^T u)}{g_a(A_a^T v)} - \frac{g_b(z)}{g_b(A_b^T v)}\right| \leq \epsilon_a \;\middle|\; A_b^T u = z\right) \mathsf{Pr}\left(A_b^T u = z\right) dz.$$

We will bound the conditional probability term. As in the proof of Theorem 5.6.5, applying Taylor's theorem with remainder (Theorem 5.3.5) gives

$$g_a(A_a^T u) = i^d \sum_{l=d}^{k_a} \mathrm{cum}_l(s_a) \frac{(i(A_a^T u))^{l-d}}{(l-d)!} + R_{k_a+1}(A_a^T u) \frac{(A_a^T u)^{k_a-d+1}}{(k_a-d+1)!}.$$

Truncating g_a after the degree $(k_a - d)$ term yields polynomial $p_a(A_a^T u)$. Denote the truncation error by $\rho_a(A_a^T u)$.

Then setting $h = \frac{g_b(A_b^T u) g_a(A_a^T v)}{g_b(A_b^T v)}$ which is a constant under our conditioning, we have

$$\begin{aligned}
\left| \frac{g_a(A_a^T u)}{g_a(A_a^T v)} - \frac{g_b(A_b^T u)}{g_b(A_b^T v)} \right| &= \frac{1}{|g_a(A_a^T v)|} \left| g_a(A_a^T u) - \frac{g_b(A_b^T u) g_a(A_a^T v)}{g_b(A_b^T v)} \right| \\
&= \frac{1}{|g_a(A_a^T v)|} \left| g_a(A_a^T u) - h \right| \\
&= \frac{1}{|g_a(A_a^T v)|} \left| p_a(A_a^T u) + \rho_a(A_a^T u) - h \right| \\
&\geq \frac{1}{|g_a(A_a^T v)|} \left| p_a(A_a^T u) - h \right| - \frac{1}{|g_a(A_a^T v)|} \left| \rho_a(A_a^T u) \right|.
\end{aligned}$$

Now we are going to show that the first term above is likely to be large and the second one is likely to be small.

We have $A_a^T u = \langle A_a, A_b \rangle A_b^T u + r^T u$ where r is a vector orthogonal to A_b. Our hypothesis, namely $1 - \langle A_a, A_b \rangle^2 \geq L^2$, gives $\|r\|^2 \geq L^2$. The orthogonality of r and A_b implies that the univariate Gaussian $r^T u$ is independent of $A_b^T u$.

Now we apply our anti-concentration inequality from Theorem 3.1.5 to obtain (for $u \sim N(0, \sigma^2 I_n)$ and fixed v satisfying (35))

$$\begin{aligned}
\mathsf{Pr}\left(\left| p_a(A_a^T u) - h \right| \leq \epsilon_a \mid A_b^T u = z \right) &\leq \frac{4(k_a - d)}{\sigma \|r\| \sqrt{2\pi}} \left(\frac{\epsilon_a (k_a - d)!}{|\mathrm{cum}_{k_a}(s_a)|} \right)^{1/(k_a - d)} \\
&\leq \frac{4(k_a - d)}{\sigma L \sqrt{2\pi}} \left(\frac{\epsilon_a (k_a - d)!}{\Delta} \right)^{1/(k_a - d)}. \qquad (38)
\end{aligned}$$

We choose

$$\epsilon_a := \frac{\Delta}{(k_a - d)!} \left(\frac{\sigma L q \sqrt{2\pi}}{4(k_a - d)} \right)^{k_a - d} \geq \frac{\Delta}{(k - d)!} \left(\frac{\sigma L q \sqrt{2\pi}}{4(k - d)} \right)^{k - d} =: \epsilon,$$

making RHS of (38) equal to q. Recall that this was for fixed v satisfying (35).

For the truncation error, assuming that the event $\left| A_a^T u \right| \leq \sigma \sqrt{2 \log 1/q}$ happens, we have

$$\begin{aligned}
\left| \rho_a(A_a^T u) \right| &\leq \left| \psi^{(k_a + 1)}(A_a^T u) \right| \cdot \frac{(A_a^T u)^{k_a - d + 1}}{(k_a - d + 1)!} \\
&\leq \frac{2^{k_a} M_{k_a + 1}}{(3/4)^{k_a + 1}} \cdot \frac{k_a!}{(k_a - d + 1)!} \cdot \left(\sigma \sqrt{2 \log 1/q} \right)^{k_a - d + 1} \\
&\leq \epsilon_a / 2,
\end{aligned}$$

where to upper bound $\left|\psi^{(k_a+1)}(A_a^T u)\right|$ we used Lemma 5.3.6, Lemma 5.3.4, and the condition $\sigma\sqrt{2\log(1/q)} \leq \frac{1}{2\sqrt{M_2}}$, which holds given our upper bound on σ. And the final inequality follows from our condition $\sigma \leq \sigma'$.

Thus with probability at least $1 - \frac{2q}{\sqrt{\pi \log 1/q}} - q$ we have $\left|p_a(A_a^T u) - h\right| - \left|\rho_a(A_a^T u)\right| \geq \epsilon_a/2$ under the condtion that $A_b^T u = z$ and v fixed. Now since this holds for any z and any fixing of v, it also holds without the conditioning on the event $A_b^T u = z$ and fixing of v.

Hence using (37), with probability at least $1 - \frac{2q}{\sqrt{\pi \log 1/q}} - q \geq 1 - 3q$ we have

$$\frac{1}{|g_a(A_a^T v)|}\left(\left|p_a(A_a^T u) - h\right| - \left|\rho_a(A_a^T u)\right|\right) \geq \epsilon_a \cdot (3/8)^d \frac{1}{(d-1)! M_d} \geq \epsilon \cdot (3/8)^d \frac{1}{(d-1)! M_d}.$$

To summarize, with probability at least $1 - 3q$ the spacing is at least ϵ. By the union bound, with probability at least $1 - 3\binom{m}{2}q$ all the spacings are at least ϵ.

□

The following is a straightforward corollary of the proof of the previous theorem.

Corollary 5.6.6. *In the setting of Theorem 5.6.5 we have with probability at least* $1 - 6mq$ *that*

$$\left|g_a(A^T u)\right|, \left|g_a(A^T v)\right| \geq \frac{\Delta_0}{2(k-d)!}\left(\frac{\sigma q\sqrt{2\pi}}{4(k-d)}\right)^{k-d}$$

for all $a \in [m]$.

An important part of the proof is to give a lower bound on the quantity $1 - \langle A_i, A_j\rangle^2 \geq L^2$ so that the ICA model remains identifiable. At order d, we will give our bounds in terms of $\sigma_m\left(A^{\odot j}\right)$ for $j = 1, \ldots, d/2$.

Lemma 5.6.7. *Fix* $m, n \in \mathbb{N}$ *such that* $n \leq m$. *Let* $A \in \mathbb{C}^{n \times m}$ *be a full row rank matrix whose columns* A_i *have unit norm. Then*

$$1 - \langle A_i, A_j\rangle^2 \geq \frac{2}{k}\sigma_m\left(A^{\odot k}\right)^2,$$

for all $k \in \mathbb{N}$ *where* $k \geq 2$.

Proof. Consider the matrix $B = A^{\odot 2}$, observe that $\langle A_i, A_j\rangle^2 = \langle B_i, B_j\rangle$. Define the matrix $C = A^{\odot k}$. Observe that $\|C_i\| = \|A_i\|^k = 1$ and that

$$1 - \langle B_i, B_j\rangle = 1 - |\langle C_i, C_j\rangle|^{2/k} \tag{39}$$

for $k \geq 2$. Recall that

$$\sigma_m(C) = \min_{\|x\|=1} \|Cx\| .$$

In particular, if we consider the vector $x = \frac{1}{\sqrt{2}}(e_i \pm e_j)$ we have

$$\|Cx\|^2 = \frac{1}{2}\left(\|C_i\|^2 + \|C_j\|^2 \pm 2\langle C_i, C_j\rangle\right) = 1 \pm \langle C_i, C_j\rangle \geq \sigma_m(C)^2.$$

Hence we must have $1 - |\langle C_i, C_j\rangle| \geq \sigma_m^2(C)$. Combining this with (39), we obtain

$$\begin{aligned} 1 - \langle B_i, B_j\rangle &= 1 - |\langle C_i, C_j\rangle|^{2/k} \\ &\geq 1 - (1 - \sigma_m(C)^2)^{2/k} \\ &\geq \frac{2}{k}\sigma_m(C)^2, \end{aligned}$$

where the last step follows from noting that all derivatives of the function $f(x) = (1-x)^t$ for $t \in (0,1)$ are negative in the interval $x \in [0,1]$ □

5.6.5 Proof of main theorem

To understand the complexity of our algorithm, we must bound how many samples are needed to estimate the matrices M_u and M_v accurately. Throughout this section, we estimate $\mathbb{E}(f(x))$ for some function $f(x)$, using independent samples x^i via

$$\bar{E}(f(x)) := \frac{1}{N}\sum_{i=1}^{N} f(x^i) \to \mathbb{E}(f(x)) .$$

More generally, we will estimate derivative tensors as follows. As before, $\phi(t) = \mathbb{E}\left(e^{it^T s}\right)$ and define the empirical version of the characteristic function in the natural way $\bar{\phi}(t) := \frac{1}{N}\sum_{i=1}^{N} e^{it^T s^i}$. As we will see, for a multiset $S \subseteq [m]$ the derivative of $\bar{\phi}(t)$ behaves nicely and will serve as an approximation of $\phi(t)$. Note that

$$\partial_S \bar{\phi}(t) = \bar{\mathbb{E}}\left(s_S e^{it^T s}\right),$$

where $\bar{\mathbb{E}}(\cdot)$ denotes empirical expectation over N i.i.d. samples of s. Similarly, we estimate $\partial_S \log \phi(t)$ by

$$\partial_S \log \bar{\phi}(t) = \bar{N}_d(t)/\bar{\phi}(t)^d, \tag{40}$$

where by Claim 5.6.4 $\bar{N}_d(t)$ is a sum of the form $\sum_{S_1,\ldots,S_d} C_{S_1,\ldots,S_d}(\partial_{S_1}\bar{\phi}(t))\ldots(\partial_{S_d}\bar{\phi}(t))$, as described in Claim 5.6.4. Thus to show that $\partial_S\bar{\phi}(t)$ is a good approximation of $\partial_S\phi(t)$ we show that $\left|\frac{N_d(t)}{\phi(t)^d} - \frac{\bar{N}_d(t)}{\bar{\phi}(t)^d}\right| = \frac{\left|\bar{\phi}(t)^d N_d(t) - \phi(t)\bar{N}_d(t)\right|}{\phi(t)^d\bar{\phi}(t)^d}$ is small.

The notion of empirical estimate of a derivative tensor now follows immediately from (40) which gives how to estimate individual entries of the tensor. Before we give our explicit bound on the sample complexity, we first need to develop a technical lemma:

Lemma 5.6.8. *Let $a_1,\ldots,a_d,b_1,\ldots,b_d \in \mathbb{C}$ be such that $|a_j - b_j| \leq \epsilon$ for real $\epsilon \geq 0$, and $|a_j| \leq R$ for $R > 0$. Then*

$$\left|\prod_{j=1}^{d} a_j - \prod_{j=1}^{d} b_j\right| \leq (R+\epsilon)^d - R^d.$$

Proof. For $0 < j < d$, define the jth elementary symmetric function in d variables: $\sigma_j(x_1,\ldots,x_d) = \sum_{1\leq i_1\leq\ldots\leq i_j\leq d} x_{i_1}\ldots x_{i_j}$. We will use the following well-known inequality (see, e.g., [122]) which holds for $x_\ell \geq 0$ for all ℓ.

$$\left(\frac{\sigma_j(x_1,\ldots,x_d)}{\binom{d}{j}}\right)^{1/j} \leq \frac{\sigma_1(x_1,\ldots,x_d)}{d}. \tag{41}$$

Let $b_j = a_j + \epsilon_j$. Then

$$\begin{aligned}\left|\prod_j (a_j+\epsilon_j) - \prod_j a_j\right| &\leq \epsilon\,\sigma_{d-1}(|a_1|,\ldots,|a_d|) + \epsilon^2\sigma_{d-2}(|a_1|,\ldots,|a_d|) + \ldots + \epsilon^{d-1}\sigma_1(|a_1|,\ldots,|a_d|)\\ &\leq d\epsilon R^{d-1} + \binom{d}{2}\epsilon^2 R^{d-2} + \ldots + \epsilon^d\\ &= (R+\epsilon)^d - R^d,\end{aligned}$$

where the second inequality follows from (41). □

With this lemma in hand, we can now give the sample complexity for our algorithm.

Lemma 5.6.9. *Let $s \in \mathbb{R}^m$ be a random vector with independent components. For $t \in \mathbb{R}^m$ let $\psi_s(t) = \phi_s(t) = \log \mathbb{E}\left(e^{it^T s}\right)$ be the second characteristic function of s. Consider the dth derivative tensor of $\psi_s(t)$ (it contains m^d entries). Let $M_2, M_{2d} > 0$ be such that $\mathbb{E}\left(s_i^2\right) \leq M_2$ and $\mathbb{E}\left(|s_i|^{2d}\right) \leq M_{2d}$. Fix $0 < \epsilon, \delta < 1/4$, and let $\|t\| \leq \frac{1}{\sqrt{2M_2}}$. Suppose we take N samples then with probability at least*

$$1 - \binom{m+d-1}{d} \frac{2^d M_{2d}}{\epsilon^2 \delta} \left[\frac{2^d d (M_d + 2)^{d-1} (d-1)!}{(3/4)^d (1/2)^d}\right]^2,$$

every entry of the empirical tensor will be within ϵ of the corresponding entry of the derivative tensor.

Proof. In light of (40) we will prove that each term in the expression for $\bar{N}_d(t)$ (it's a product of several $\partial_S \bar{\phi}(t)$) is a good approximation of the corresponding term in the expression for $N_d(t)$ by showing that the corresponding factors in the product are close. Finally, we show that the whole sum is a good approximation. For complex-valued r.v. X with mean μ, note that $\mathrm{Var}(X) = \mathbb{E}\left((X-\mu)(X-\mu)^*\right) = \mathbb{E}\left(XX^*\right) - \mu\mu^* \leq \mathbb{E}\left(|X|^2\right)$.

We use the second moment method to prove that $\partial_S \bar{\phi}(t)$ is a good approximation of $\partial_S \phi(t)$ with high probability.

For multiset $S \subseteq \{i_1, \ldots, i_d\}$, with p_j being the frequency of j in S, by the same arguments as in (33) we have

$$\mathrm{Var}\left(s_S e^{it^T s}\right) \leq \mathbb{E}\left(s_S^2\right) \leq \prod_{j=1}^{m} \mathbb{E}\left(s_j^{2p_j}\right) \leq \prod_{j=1}^{m} \left(\mathbb{E}\left(s_j^{2d}\right)\right)^{p_j/d} \leq \left(\max_j \mathbb{E}\left(s_j^{2d}\right)\right)^{|S|/d} \leq M_{2d}^{|S|/d}.$$

Thus

$$\mathrm{Var}\left(\partial_S \bar{\phi}(t)\right) \leq \frac{M_{2d}^{|S|/d}}{N} \leq \frac{M_{2d}}{N}.$$

Chebyshev's inequality (which remains unchanged for complex-valued r.v.s) for $\epsilon' > 0$ yields

$$\mathsf{Pr}\left(\left|\partial_S \bar{\phi}(t) - \partial_S \phi(t)\right| \geq \epsilon'\right) \leq \frac{M_{2d}}{\epsilon'^2 N}. \tag{42}$$

We will choose a value of ϵ' shortly.

Now we bound the difference between the corresponding summands in the decompositions of $N_d(t)$ and $\bar{N}_d(t)$ as sums. Specifically, with probability at most $\frac{(d+1)M_{2d}}{\epsilon' N}$ (this comes from the union bound: we want the event in (42) to hold for all S_j for $j \in [d]$ as well as for $S = \emptyset$, corresponding to $\phi(t)$) we have

$$\begin{aligned}
\left|\left(\bar{\phi}(t)^d \prod_{j=1}^{d} \partial_{S_j}\phi(t)\right) - \left(\phi(t)^d \prod_{j=1}^{d} \partial_{S_j}\bar{\phi}(t)\right)\right| &\leq \left|\prod_{j=1}^{d} \partial_{S_j}\phi(t)\right| \left|\bar{\phi}(t)^d - \phi(t)^d\right| + \left|\phi(t)^d\right| \left|\prod_{j=1}^{d} \partial_{S_j}\phi(t) - \prod_{j=1}^{d} \partial_{S_j}\bar{\phi}\right. \\
&\leq M_d \left|\bar{\phi}(t)^d - \phi(t)^d\right| + (M_d + \epsilon')^d - M_d^d \\
&\leq \epsilon' d\, M_d + \epsilon' d (M_d + \epsilon')^{d-1} \\
&\leq 2\epsilon' d (M_d + 1 + \epsilon')^{d-1},
\end{aligned}$$

where the second inequality used (33), Lemma 5.6.8 and $|\phi(t)| \leq 1$ and $\left|\bar{\phi}(t)\right| \leq 1$.

Now using the expression for $N_d(t)$ as a sum given in Claim 5.6.4, with probability at most $\frac{2^d M_{2d}}{\epsilon'^2 N}$ (the factor 2^d again comes from the union bound: we want the event in (42) to hold for all (multi-) subsets of $\{i_1, \ldots, i_d\}$) we have

$$\begin{aligned}
\left|\partial_S \bar{\phi}(t) - \partial_S \phi(t)\right| &= \frac{\left|\bar{\phi}(t)^d N_d(t) - \phi(t)\bar{N}_d(t)\right|}{\left|\phi(t)^d \bar{\phi}(t)^d\right|} && (43) \\
&\leq \frac{2^d \epsilon' d (M_d + 1 + \epsilon')^{d-1}(d-1)!}{\left|\phi(t)^d \bar{\phi}(t)^d\right|} \\
&\leq \frac{2^d \epsilon' d (M_d + 1 + \epsilon')^{d-1}(d-1)!}{(3/4)^d (3/4 - \epsilon')^d}, && (44) \\
&\leq \epsilon,
\end{aligned}$$

where the last inequality used Lemma 5.3.4 and

$$\epsilon' = \left[\frac{2^d d (M_d + 1 + \epsilon')^{d-1}(d-1)!}{(3/4)^d (3/4 - \epsilon')^d}\right]^{-1} \epsilon.$$

Now if we want (44) to hold for all multisets S of size d, then the union bound needs to extended to all such multisets (of which there are $\binom{m+d-1}{d}$) giving that error probability at most

$$\binom{m+d-1}{d} \frac{2^d M_{2d}}{\epsilon'^2 N} = \binom{m+d-1}{d} \frac{2^d M_{2d}}{\epsilon^2 N} \left[\frac{2^d d (M_d + 1 + \epsilon')^{d-1}(d-1)!}{(3/4)^d (3/4 - \epsilon')^d}\right]^2,$$

as desired. □

Lemma 5.6.10 (Sample Complexity). *Let $x = As$ be an ICA model with $A \in \mathbb{R}^{n\times m}, x \in \mathbb{R}^n, s \in \mathbb{R}^m$ and d an even positive integer. Let $M_2, M_{2d} > 0$ be such that $\mathbb{E}\left(s_i^2\right) \leq M_2$ and $\mathbb{E}\left(|s_i|^{2d}\right) \leq M_{2d}$. Let $v \in \mathbb{R}^n$ satisfy $\|v\|_2 \leq \frac{1}{2\|A\|_2\sqrt{2M_2}}$. Let $T_v = D_u^d\psi_x(v)$ be the d'th derivative tensor of $\psi_x(u) = \log \mathbb{E}\left(e^{iu^Tx}\right)$ at v. And let $\bar{T}_v = D_u^d\bar{\psi}_x(v)$ be its naive estimate using N independent samples of x where*

$$N \geq \binom{m+d-1}{d} \frac{1}{m^{d/2}\sigma_1(A)^d} \frac{2^d M_{2d}}{\epsilon^2\delta} \left[\frac{2^d d(M_d+2)^{d-1}(d-1)!}{(3/4)^d(1/2)^d}\right]^2.$$

Then with probability at least $1-\delta$ we have

$$\left\|T_v - \bar{T}_v\right\|_F \leq \epsilon.$$

Proof. In the following all tensors are flattened into matrices. Let $x^j = As^j$, $j \in [N]$ be i.i.d. samples. Letting $t = A^Tv$ we have $T_v = D_u^d\psi_x(u) = A^{\otimes d/2}D_t^d\psi_s(t)(A^{\otimes d/2})^T$, and $\bar{T}_v = D_u^d\bar{\psi}_x(u) = A^{\otimes d/2}D_t^d\bar{\psi}_s(t)(A^{\otimes d/2})^T$. (Note that we could also have written $D_u^d\psi_x(u) = A^{\odot d/2}\mathrm{diag}\left(\partial_{t_j}^d\psi_s(t)\right)(A^{\odot d/2})^T$ because the components of s are indpendent, however the corresponding empirical equation $D_u^d\bar{\psi}_x(u) = A^{\odot d/2}\mathrm{diag}\left(\partial_{t_j}^d\bar{\psi}_s(t)\right)(A^{\odot d/2})^T$ need not be true.)

Hence

$$\begin{aligned}
\left\|\bar{T}_v - T_v\right\|_F &= \left\|A^{\otimes d/2}D_t^d\bar{\psi}_s(t)(A^{\otimes d/2})^T - A^{\otimes d/2}D_t^d\psi_s(t)(A^{\otimes d/2})^T\right\|_F \\
&= \left\|A^{\otimes d/2}(D_t^d\bar{\psi}_s(t) - D_t^d\psi_s(t))(A^{\otimes d/2})^T\right\|_F \\
&\leq \sigma_1(A^{\otimes d/2})^2\left\|D_t^d\bar{\psi}_s(t) - D_t^d\psi_s(t)\right\|_F \\
&= \sigma_1(A)^d\left\|D_t^d\bar{\psi}_s(t) - D_t^d\psi_s(t)\right\|_F \\
&\leq \epsilon,
\end{aligned}$$

where the last inequality holds with probability at least $1-\delta$ by Lemma 5.6.9 which is applicable because $\left\|A^Tv\right\|_2 \leq \|A\|_2\|v\|_2 \leq \frac{1}{2\sqrt{2M_2}}$. □

We are now ready to formally state and prove the main theorem. To get a success probability of 3/4, we choose q so that $20m^2q < 1/4$.

Theorem 5.6.11 (Underdetermined ICA)**.** *Let $x \in \mathbb{R}^n$ be generated by an underdetermined ICA model $x = As$ with $A \in \mathbb{R}^{n\times m}$ where $n \leq m$. Suppose that the following data and conditions are given:*

1. *$d \in 2\mathbb{N}$ such that $\sigma_m\left(A^{\odot d/2}\right) > 0$.*

2. *k such that for each i there exists k_i, where $d < k_i < k$ such that $|\mathrm{cum}_{k_i}(s_i)| \geq \Delta_0$.*

3. *Constants M_2, M_d, M_k such that for each s_i the following bounds hold*

$$\mathbb{E}(s_i) = 0, \quad \mathbb{E}\left(s_i^2\right) \leq M_2, \quad \mathbb{E}\left(s_i^d\right) \leq M_d, \quad \mathbb{E}\left(|s_i|^{k_i+1}\right) \leq M_k, \quad \Delta_0 \leq M_d, \quad \mathbb{E}\left(|s_i|^{2d}\right) \leq M_{2d}.$$

4. *$0 < \sigma \leq \min(1, \sigma_0, \frac{1}{4m}\sqrt{\frac{1}{6M_2\ln(2/q)}})$ where*

$$\sigma_0 = \Delta_0 \frac{k-d+1}{k!}\left(\frac{3}{8}\right)^k \frac{1}{M_k}\left(\frac{2\sigma_m(A^{\odot d/2})q\sqrt{2\pi}}{4(k-d)\sqrt{d}}\right)^{k-d}\left(\frac{1}{\sqrt{2\log 1/q}}\right)^{k-d}.$$

*Then, with probability at least $1-20m^2q$, algorithm **Underdetermined ICA**(σ) will return a matrix $\tilde{B}$ such that there exist signs $\alpha_j \in \{-1,1\}$ and permutation $\pi : [m] \to [m]$ such that*

$$\left\|B_j - \alpha_j \tilde{B}_{\pi(j)}\right\| \leq \epsilon,$$

using N samples where

$$N \geq \left(\frac{km(M_d+2)}{\sigma q \sigma_m(A^{\odot d/2})}\right)^{ck} \frac{\kappa(A^{\odot d/2})^6 M_{2d}}{\Delta_0^6 \epsilon^2},$$

for some absolute constant c. The running time of the algorithm is poly(N).

Proof. The proof involves putting together of various results we have proven. We take N independent samples of x and form the flattened dth derivative tensors $\bar{M}_u, \bar{M}_v$ of $\psi(u)$ evaluated at u and v which are sampled from $N(0, \sigma_0^2)$. Recall that these are the matrices constructed by **Underdetermined ICA**(σ_0)) which then invokes **Diagonalize**$(\cdot)$ which computes eigendecomposition of $\bar{M}_u\bar{M}_v^{-1}$. We will denote by M_u, M_v the corresponding matrices without any sampling errors. We will first use the result about eigenvalue spacing Theorem 5.6.5 to get a bound on the spacings of the eigenvalues of the matrix $M_uM_v^{-1}$,

where $u, v \sim N(0, \sigma_0^2 I_n)$ are random vectors. Next, we determine upper and lower bounds K_U and K_L on the eigenvalues of M_u and M_v. We can then apply Theorem 4.3.5 to show that if we have sufficiently good approximation of M_u and M_v then we will get a good reconstruction of matrix A. Finally, we use Lemma 5.6.10 to determine the number of samples needed to get the required approximation.

Step 1. First, we apply Theorem 5.6.5. Note that our choice of σ_0 satisfies the constraints on σ in Theorem 5.6.5; thus except with probability q, we have

$$\left|\frac{g_b(A_b^T u)}{g_b(A_b^T v)} - \frac{g_a(A_a^T u)}{g_a(A_a^T v)}\right| \geq \Omega_0 := \frac{\Delta_0}{M_d}\left(\frac{3}{8}\right)^d \frac{1}{(d-1)!(k-d)!}\left(\frac{\sigma_0 q L\sqrt{2\pi}}{4(k-d)}\right)^{k-d}, \tag{45}$$

for all pairs $a, b \in [m]$. Here L as defined in Theorem 5.6.5 is given by $2\sigma_m(A^{\odot d/2})/\sqrt{d}$ by Lemma 5.6.7.

Step 2. Next, we will show that u and v concentrate in norm. To do so, we will apply the following concentration inequality for sub-exponential random variables (this is standard in the proof of the Johnson-Lindenstrauss Lemma, see [16, 49] or alternatively [130] for a more general formulation).

Lemma 5.6.12 ([16, 49])**.** *Let $z_i \sim N(0,1)$ be i.i.d., then*

$$\mathsf{Pr}\left(\sum_{i=1}^n z_i^2 \geq \beta n\right) \leq e^{\frac{n}{2}(1-\beta+\log(\beta))}.$$

For $\beta \geq 6$, the bound only improves as n increases. Thus, we have the simplified bound

$$\mathsf{Pr}\left(\sum_{i=1}^n z_i^2 \geq \beta n\right) \leq e^{-\frac{n\beta}{12}}.$$

In particular, union bounding over both $u, v \in \mathbb{R}^n$, we have

$$\begin{aligned}\mathsf{Pr}\left(\|u\|, \|v\| \geq \frac{1}{2\|A\|_F\sqrt{2M_2}}\right) &= \mathsf{Pr}\left(\|u\|^2, \|v\|^2 \geq \left(\frac{1}{2\|A\|_F\sqrt{2M_2}}\right)^2\right) \\ &\leq 2\exp\left(-\frac{1}{12\sigma_0^2}\left(\frac{1}{2\|A\|_F\sqrt{2M_2}}\right)^2\right),\end{aligned}$$

where in the second line, we took $\beta n = \frac{1}{12\sigma_0^2}\left(\frac{1}{2\|A\|_F\sqrt{2M_2}}\right)^2$. Using $\|A\|_F \leq m$, and our choice of σ_0 which gives $\sigma_0 \leq \frac{1}{4m}\sqrt{\frac{1}{6M_2\ln(2/q)}}$ we obtain

$$\mathsf{Pr}\left(\|u\|, \|v\| \geq \frac{1}{2\|A\|_F\sqrt{2M_2}}\right) \leq q.$$

Thus except with probability q, norms $\|u\|, \|v\|$ satisfy the hypotheses of Lemma 5.6.10.

Step 3. Now we determine the values of parameters K_U and K_L used in Theorem 4.3.5. A bound for K_U can be obtained from Lemma 5.3.6 and Lemma 5.3.4 to $\psi_s(t) = \psi_s(A^T u)$. The latter lemma being applicable because $\left\|A^T u\right\| \leq \|A\|_F \|u\| \leq \frac{1}{2\sqrt{2M_2}}$ and $\left\|A^T v\right\| \leq \|A\|_F \|v\| \leq \frac{1}{2\sqrt{2M_2}}$ from Step 2:

$$K_U = \frac{(d-1)!2^{d-1}M_d}{(3/4)^d}.$$

For K_L, by Cor. 5.6.6 we can set

$$K_L = \frac{\Delta_0}{2(k-d)!}\left(\frac{\sigma_0 q\sqrt{2\pi}}{4(k-d)}\right)^{k-d},$$

which holds with probability at least $1 - 6mq$.

Step 4. We now fix K_1 which is the upper bound on $\left\|M_u - \bar{M}_u\right\|_F$ and $\left\|M_v - \bar{M}_v\right\|_F$ needed in Theorem 4.3.5 (the role of these two quantities is played by $\|R_\mu\|_F$ and $\|R_\lambda\|_F$ in that theorem). Our assumption $\Delta_0 \leq M_d$ gives that $\Omega_0 \leq 1$ by (45). And hence the bound required in Theorem 4.3.5 becomes

$$K_1 = \frac{\epsilon K_L^2 \sigma_m(B)^3}{2^{11}\kappa(B)^3 K_U m^2}\Omega_0, \tag{46}$$

where $B = A^{\odot d/2}$.

For this K_1 by Theorem 4.3.5 the algorithm recovers $\tilde{B}$ with the property that there are signs $\alpha_j \in \{-1, 1\}$ and permutation $[m] \to [m]$ such that

$$\left\|B_j - \alpha_j \tilde{B}_{\pi(j)}\right\| \leq \epsilon.$$

Step 5. It remains to determine the number of samples needed to achieve $\left\|M_u - \bar{M}_u\right\|_F \leq K_1$ and $\left\|M_v - \bar{M}_v\right\|_F \leq K_1$.

By Step 2 above, we satisfy the hypotheses of Lemma 5.6.10. Hence by that lemma, for N at least the quantity below

$$\binom{m+d-1}{d}\frac{1}{m^{d/2}\sigma_1(A)^d}\frac{2^d M_{2d}}{K_1^2 q}\left(\frac{16^d d(M_d+2)^{d-1}(d-1)!}{3^d}\right)^2 \leq 11^{2d} m^{d/2} d^{2(d+1)} M_{2d}(M_d+2)^{2(d-1)}\frac{1}{K_1^2 q}$$

we have

$$\left\|M_u - \bar{M}_u\right\|_F \leq K_1,$$

except with probability q, and similarly for $\left\|M_u - \bar{M}_u\right\|_F$. Subtistuting the value of K_1 from (46) and in turn of K_U, K_L and Ω_0 above and simplifying (we omit the straightforward but tedious details) gives that it suffices to take

$$N \geq \frac{2^{4k+6d+26}}{3^{2d}} d^{6d+2}(k-d)^{2(k-d)} m^{d/2+4} \frac{M_d^2 M_{2d}(M_d+2)^{2d}}{\Delta_0^6} \frac{\kappa(B)^6}{\sigma_m(B)^{k-d+6}} \frac{1}{\sigma^{5(k-d)} q^{5(k-d)+1}} \frac{1}{\epsilon^2}.$$

Accounting for the probability of all possible bad events enumerated in the proof via the union bound we see that with probability at least $1 - q - 3\binom{m}{2}q - 6mq - q > 1 - 20m^2 q$ no bad events happen. The running time computation involves empirical estimates of derivate tensors and SVD and eigenvalue computations; we skip the routine check that the running time is poly(N). □

5.6.6 Gaussian noise

Theorem 5.6.11 just proved is the detailed version of Theorem 5.1.2 without Gaussian noise. In this section we indicate how to extend this proof when there is Gaussian noise thus proving Theorem 5.1.2 in full. Our algorithm for the noiseless case applies essentially unaltered to the case when the input has unknown Gaussian noise if $d > 2$. We comment on the case $d = 2$ at the end of this section. More precisely, the ICA model now is

$$x' = x + \eta = As + \eta,$$

where $\eta \sim N(0, \Sigma)$ where $\sigma \in \mathbb{R}^{n \times n}$ is unknown covariance matrix and η is independent of s. Using the independence of η and s and the standard expression for the second characteristic of the Gaussian we have

$$\psi_{x'}(u) = \mathbb{E}\left(e^{iu^T x'}\right) = \mathbb{E}\left(e^{iu^T x + iu^T \eta}\right) = \psi_x(u) + \psi_\eta(u) = \psi_x(u) - \frac{1}{2} u^T \Sigma u. \qquad (47)$$

Our algorithm works with (estimate of) the dth derivative tensor of $\psi_{x'}(u)$. For $d > 2$, we have $D_u^d \psi_{x'}(u) = D_u^d \psi_x(u)$ as in (47) the component of the second characteristic function corresponding to the Guassian noise is quadratic and vanishes for third and higher

derivatives. Therefore, but for the estimation errors, the Gaussian noise makes no difference and the algorithm would still recover A as before. Since the algorithm works only with estimates of these derivatives, we have to account for how much our estimate of $D_u^d \psi_x(u)$ changes due to the extra additive term involving the derivative of the estimate of the second characteristic of the Guassian.

If Σ is such that the moments of the Gaussian noise also satisfy the conditions we imposed on the moments of the s_i in the Theorem 5.6.11, then we can complete the proof with little extra work. The only thing that changes in the proof of the main theorem is that instead of getting the bound $\left\|M_u - \bar{M}_u\right\| \leq \epsilon'$ we get the bound $\left\|M_u - \bar{M}_u\right\| \leq 2\epsilon'$. If we increase the number of samples by a factor of 4 then this bound becomes $\left\|M_u - \bar{M}_u\right\| \leq \epsilon'$, and so the proof can be completed without any other change.

The $d = 2$ case. When $d = 2$, the second derivative of the component of the second characteristic function corresponding to the noise in (47) is a constant matrix independent of u. Thus if we take derivatives at two different points and subtract them, then this constant matrix disappears. This is analogous to the algorithm we gave for fully-determined ICA with noise in Sec. 5.3.5. The error analysis can still proceed along the above lines; we omit the details.

CHAPTER VI

CONCLUSION AND FUTURE DIRECTIONS

In this thesis, we have designed and analysed a pair of novel tensor decompositions algorithms for certain unsupervised learning tasks. In the process, we have advanced the state of the art both in the algorithmic theory of tensor decompositions and also unsupervised learning. The highlights of our unsupervised learning results are certainly the first provably efficient underdetermined ICA algorithm and also the new model for feature selection. Our algorithms, in all cases, are efficient in terms of computational and sample complexity, and are all robust to the natural noise that arises from sampling. Below, we summarise open questions and promising directions for this research:

Give an efficient algorithm for independent subspace analysis (ISA). This is the problem where the s_i are not all indendent but rather the set of indices $[m]$ is partitioned into subsets. For any two distinct subsets S_1 and S_2 in the partition s_{S_1} is independent of s_{S_2}, where s_{S_1} denotes the vector of the s_i with $i \in S_1$ etc. Clearly this problem is a generalization of ICA, and is essentially a form of the relevant feature algorithm using tensor local search that we explore in Section 4.2. There, though, our analysis was too weak and the error accumulated rapidly (exponentially) over the iterations. A concrete approach would be to try to adapt the tensor flattening/eigenvalue decomposition for general matrices ansatz to the problem. Recall that all our applications of the rank 1 decomposition require the delocalisation of ensembles of certain random variables. This becomes dramatically more difficult when the ensemble is not independent as in the ISA case due to the block structure. Thus achieving the delocalisation of eigenvalues of a block where there is only limited randomness is the challenge here.

Can the tensor decomposition approach give an algorithm that can solve the Gaussian mixture model in full generality? That is to say, can we separate $k > n$ Gaussians distributions with arbitrary mean and variance parameters. Already, the work of [13] is able to

deal with $k > n$ spherical Gaussians, but the "Poissonisation" technique of that paper does not seem to extend to the general variance case. What other techniques are necessary to solve this problem?

Our condition for ICA to be possible required that there exist a d such that $A^{\odot d}$ has full column rank. As mentioned before, the existence of such a d turns out to be equivalent to the necessary and sufficient condition for ICA, namely, any two columns of A are linearly independent. Thus if d is large for a matrix A then our algorithm whose running time is exponential in d will be inefficient. This is inevitable to some extent as suggested by the ICA lower bound in [13]. However, the lower bound there requires that one of the s_i be Gaussian. Can one prove the lower bound without this requirement?

Our recursive partitioning algorithm for fully determined ICA is extremely efficient practically, in terms of computational and sample requirements – almost unbelievably so. On the other hand, our underdetermined algorithm has far inferior practical performance even when the matrix A is of size $n \times (n+1)$ which really is a minimal underdetermined example. Giving a practically efficient algorithm for this case would be a very interesting challenge. In particular, it is tempting to go after a recursive partitioning type idea here once again, but it is not clear what the subproblems are, and how to recurse on them. But this surely must be the right way to proceed.

The two notions of tensor decomposition we study in this thesis both fail to generalise the spectral decomposition for symmetric, Hermitian or normal matrices properly. Specifically, the decomposition in Section 4.3 captures the idea of low-rank decomposition, and the decomposition in Section 4.2 captures the variational characterisation of eigenvalues. From a theoretical perspective, it would be interesting to try to unify these two ideas and to do so in a computationally tractable way. Such an algorithm would be rather powerful and would no doubt find many applications. The theory here is very weak – for example, the spectral theorem applies for Hermitian operators over any (potentially infinite dimensional) Hilbert space, analytic issues of convergence aside. Perhaps, this is the right way forward to obtain a good theory – leave aside the temptations of finite dimension and deal with harder, less structured spaces which will reveal the essential structure of the problem.

A second problem in this area is to try to find further applications of the tensor decompositions that we've described. For the pairwise tensor decomposition, we already can tackle underdetermined ICA and fully determined mixture of spherical Gaussians. What other problems are susceptible to this approach? Is their any gain to be had in viewing decompositions as some form of kernel PCA? Are there tensor decomposition techniques that are not fundamentally linear? What does one do for tensors of order 3 where neither the flattening nor inserting a random argument makes any progress on this problem. What techniques are needed to solve this problem?

For the majority of Section 5.2, we did not dwell on learning geometrically robust concepts as of more interest to us was the unsupervised dimensionality reduction step. On the other hand, these geometrically robust concepts might be a good model for learning under noise in that the noise model is not too strong (i.e., agnostic learning which allows arbitrary noise) and the noise model is informed by the concept class that we're trying to learn (i.e., the errors naturally should occur only along the decision boundary and not deep inside the two binary classes). A natural question is what can one learn efficiently in such a model? In our context, we assume that the dimensionality is quite low once we're in the relevant subspace and can apply naive VC bounds, but one should be able to do much better.

REFERENCES

[1] "The FastICA package for MATLAB." `http://research.ics.aalto.fi/ica/fastica/`. Accessed: 2014-08-20.

[2] Adamczak, R., Litvak, A., Pajor, A., and Tomczak-Jaegermann, N., "Quantitative estimates of the convergence of the empirical covariance matrix in logconcave ensembles," *J. Amer. Math. Soc.*, vol. 233, pp. 535–561, 2011.

[3] Agarwal, A., Anandkumar, A., Jain, P., Netrapalli, P., and Tandon, R., "Learning sparsely used overcomplete dictionaries via alternating minimization," *arXiv preprint arXiv:1310.7991*, 2013.

[4] Albera, L., Ferréol, A., Comon, P., and Chevalier, P., "Blind identification of overcomplete mixtures of sources (biome)," *Linear algebra and its applications*, vol. 391, pp. 3–30, 2004.

[5] Alon, N., Andoni, A., Kaufman, T., Matulef, K., Rubinfeld, R., and Xie, N., "Testing k-wise and almost k-wise independence," in *Proceedings of STOC*, pp. 496–505, 2007.

[6] Alon, N., Krivelevich, M., and Sudakov, B., "Finding a large hidden clique in a random graph," in *SODA*, pp. 594–598, 1998.

[7] Ames, B. P. W. and Vavasis, S. A., "Nuclear norm minimization for the planted clique and biclique problems," *Math. Program.*, vol. 129, no. 1, pp. 69–89, 2011.

[8] Anandkumar, A., Foster, D., Hsu, D., Kakade, S., and Liu, Y.-K., "A spectral algorithm for latent dirichlet allocation," in *Advances in Neural Information Processing Systems 25*, pp. 926–934, 2012.

[9] Anandkumar, A., Ge, R., Hsu, D., Kakade, S. M., and Telgarsky, M., "Tensor decompositions for learning latent variable models," *CoRR*, vol. abs/1210.7559, 2012.

[10] Anandkumar, A., Ge, R., and Janzamin, M., "Provable learning of overcomplete latent variable models: Semi-supervised and unsupervised settings," *arXiv preprint arXiv:1408.0553*, 2014.

[11] Anandkumar, A., Hsu, D., and Kakade, S. M., "A method of moments for mixture models and hidden markov models," in *Proc. of COLT*, 2012.

[12] Anderson, J., Goyal, N., and Rademacher, L., "Efficient learning of simplices," *COLT*, 2013.

[13] Anderson, J., Belkin, M., Goyal, N., Rademacher, L., and Voss, J., "The more, the merrier: the blessing of dimensionality for learning large gaussian mixtures," *arXiv:1311.2891*, 2013.

[14] Arora, S., Ge, R., and Moitra, A., "New algorithms for learning incoherent and overcomplete dictionaries," in *Proceedings of The 27th Conference on Learning Theory*, pp. 779–806, 2014.

[15] Arora, S., Ge, R., Moitra, A., and Sachdeva, S., "Provable ICA with unknown gaussian noise, with implications for gaussian mixtures and autoencoders," in *NIPS*, pp. 2384–2392, 2012.

[16] Arriaga, R. I. and Vempala, S., "An algorithmic theory of learning: Robust concepts and random projection," *Machine Learning*, vol. 63, no. 2, pp. 161–182, 2006.

[17] Bar-Ness, J. W., Carlin, Y., and Steinberger, M. L., "Bootstrapping adaptive interference cancelers - some practical limitations," in *Proc. the Globecom Conference*, pp. 1251–1255, 1982.

[18] Barak, B., Kelner, J. A., and Steurer, D., "Dictionary learning and tensor decomposition via the sum-of-squares method," *arXiv preprint arXiv:1407.1543*, 2014.

[19] Bauer, F. L. and Fike, C., "Norms and exclusion theorems," *Numerische Mathematik*, vol. 2, no. 1, pp. 137–141, 1960.

[20] Baum, E. B., "On learning a union of half spaces," *J. Complexity*, vol. 6, no. 1, pp. 67–101, 1990.

[21] Belkin, M., Rademacher, L., and Voss, J., "Blind signal separation in the presence of Gaussian noise," in *Proc. of COLT*, 2013.

[22] Belkin, M. and Sinha, K., "Polynomial learning of distribution families," in *FOCS*, pp. 103–112, 2010.

[23] Belkin, M. and Sinha, K., "Toward learning gaussian mixtures with arbitrary separation," in *COLT*, pp. 407–419, 2010.

[24] Bell, A. J. and Sejnowski, T. J., "An information-maximization approach to blind separation and blind deconvolution," *Neural Comput.*, vol. 7, pp. 1129–1159, Nov. 1995.

[25] Bhaskara, A., Charikar, M., Moitra, A., and Vijayaraghavan, A., "Smoothed analysis of tensor decompositions," *CoRR*, vol. abs/1311.3651, 2013.

[26] Bhatia, R., *Matrix analysis*, vol. 169. Springer, 1997.

[27] Blum, A. and Kannan, R., "Learning an intersection of k halfspaces over a uniform distribution," in *FOCS*, pp. 312–320, 1993.

[28] Blum, A. and Kannan, R., "Learning an intersection of a constant number of halfspaces over a uniform distribution," *J. Comput. Syst. Sci.*, vol. 54, no. 2, pp. 371–380, 1997.

[29] Blum, A. L., "Relevant examples and relevant features: Thoughts from computational learning theory," in *AAAI Fall Symposium on 'Relevance'*, 1994.

[30] Blumer, A., Ehrenfeucht, A., Haussler, D., and Warmuth, M. K., "Learnability and the varpnik-chervovenkis dimension," *Journal of the ACM*, vol. 6, pp. 929–965, 1989.

[31] Borwein, P. and Erdelyi, T., *Polynomials and Polynomial Inequalities*. Springer, 1995.

[32] Brubaker, S. and Vempala, S., "Random tensors and planted cliques," *Approximation, Randomization, and Combinatorial Optimization. Algorithms and Techniques*, vol. 5687, pp. 406–419, 2009.

[33] Brubaker, S. C. and Vempala, S., *Extensions of Principal Component Analysis*. PhD thesis, Georgia Institute of Technology, 2009.

[34] Brubaker, S. C. and Vempala, S. S., "Isotropic PCA and affine-invariant clustering," in *Building Bridges*, pp. 241–281, Springer, 2008.

[35] Carbery, A. and Wright, J., "Distributional and L^q norm inequalities for polynomials over convex bodies in R^n," *Mathematical Research Letters*, vol. 8, pp. 233–248, 2001.

[36] Cardoso, J.-F. and Laheld, B., "Equivariant adaptive source separation," *Signal Processing, IEEE Transactions on*, vol. 44, pp. 3017 –3030, dec 1996.

[37] Cardoso, J., "Source separation using higher order moments," in *International Conference on Acoustics, Speech, and Signal Processing*, 1989.

[38] Cardoso, J., "Super-symmetric decomposition of the fourth-order cumulant tensor. blind identification of more sources than sensors," in *Acoustics, Speech, and Signal Processing, 1991. ICASSP-91., 1991 International Conference on*, pp. 3109–3112, IEEE, 1991.

[39] Carroll, J. D. and Chang, J.-J., "Analysis of individual differences in multidimensional scaling via an n-way generalization of Eckart-Young decomposition," *Psychometrika*, vol. 35, no. 3, pp. 283–319, 1970.

[40] Chang, J. T., "Full reconstruction of markov models on evolutionary trees: identifiability and consistency," *Mathematical biosciences*, vol. 137, no. 1, pp. 51–73, 1996.

[41] Chaudhuri, K. and Rao, S., "Learning mixtures of product distributions using correlations and independence," in *Proc. of COLT*, 2008.

[42] Cichocki, A., Amari, S., Siwek, K., Tanaka, T., and Phan, A., "ICALAB toolboxes." `http://www.bsp.brain.riken.jp/ICALAB/`. Accessed: 2014-08-20.

[43] Coja-Oghlan, A., "Graph partitioning via adaptive spectral techniques," *Combinatorics, Probability & Computing*, vol. 19, no. 2, pp. 227–284, 2010.

[44] Comon, P., "Independent Component Analysis," in *Proc. Int. Sig. Proc. Workshop on Higher-Order Statistics*, (Chamrousse, France), pp. 111–120, July 10-12 1991. Keynote address. Republished in *Higher-Order Statistics*, J.L.Lacoume ed., Elsevier, 1992, pp 29–38.

[45] Comon, P., "Independent Component Analysis, a new concept ?," *Signal Processing, Elsevier*, vol. 36, pp. 287–314, Apr. 1994. Special issue on Higher-Order Statistics. hal-00417283.

[46] Comon, P. and Rajih, M., "Blind identification of under-determined mixtures based on the characteristic function," *Signal Processing*, vol. 86, no. 9, pp. 2271–2281, 2006.

[47] Comon, P. and Jutten, C., eds., *Handbook of Blind Source Separation.* Academic Press, 2010.

[48] Dasgupta, S., "Learning mixtures of Gaussians," in *Foundations of Computer Science, 1999. 40th Annual Symposium on*, pp. 634–644, IEEE, 1999.

[49] Dasgupta, S. and Gupta, A., "An elementary proof of a theorem of johnson and lindenstrauss," *Random Structures and Algorithms*, vol. 22, no. 1, pp. 60–65, 2003.

[50] Dasgupta, S. and Schulman, L., "A probabilistic analysis of EM for mixtures of separated, spherical Gaussians," *The Journal of Machine Learning Research*, vol. 8, pp. 203–226, 2007.

[51] Davis, C. and Kahan, W. M., "The rotation of eigenvectors by a perturbation III," *SIAM Journal on Numerical Analysis*, vol. 7, no. 1, pp. 1–46, 1970.

[52] De Lathauwer, L., Castaing, J., and Cardoso, J., "Fourth-order cumulant-based blind identification of underdetermined mixtures," *Signal Processing, IEEE Transactions on*, vol. 55, no. 6, pp. 2965–2973, 2007.

[53] De Lathauwer, L., De Moor, B., and Vandewalle, J., "On the best rank-1 and rank-$(R_1, R_2, \ldots, R_n)$ approximation of higher-order tensors," *SIAM Journal on Matrix Analysis and Applications*, vol. 21, no. 4, pp. 1324–1342, 2000.

[54] Dekel, Y., Gurel-Gurevich, O., and Peres, Y., "Finding hidden cliques in linear time with high probability," in *Proceedings of ANALCO*, pp. 67–75, 2011.

[55] Delfosse, N. and Loubaton, P., "Adaptive blind separation of independent sources: A deflation approach," *Signal Processing*, vol. 45, no. 1, pp. 59 – 83, 1995.

[56] Eisenstat, S. C. and Ipsen, I. C., "Relative perturbation results for eigenvalues and eigenvectors of diagonalisable matrices," *BIT Numerical Mathematics*, vol. 38, no. 3, pp. 502–509, 1998.

[57] Feige, U. and Krauthgamer, R., "Finding and certifying a large hidden clique in a semirandom graph," *Random Struct. Algorithms*, vol. 16, no. 2, pp. 195–208, 2000.

[58] Feige, U. and Ron, D., "Finding hidden cliques in linear time," in *Proceedings of AofA*, pp. 189–204, 2010.

[59] Feige, U. and Krauthgamer, R., "The probable value of the Lovász–Schrijver relaxations for maximum independent set," *SICOMP*, vol. 32, no. 2, pp. 345–370, 2003.

[60] Feldman, V., Grigorescu, E., Reyzin, L., Vempala, S., and Xiao, Y., "Statistical algorithms and a lower bound for planted clique," in *STOC*, 2013.

[61] Feller, W., *An Introduction to Probability Theory and its Applications, vol 1.* John Wiley and sons, 1968.

[62] Frieze, A. M., Jerrum, M., and Kannan, R., "Learning linear transformations," in *FOCS*, pp. 359–368, 1996.

[63] Frieze, A. M. and Kannan, R., "A new approach to the planted clique problem," in *FSTTCS*, pp. 187–198, 2008.

[64] Golub, G. H. and Loan, C. F. V., *Matrix Computations.* The Johns Hopkins University Press, 2013.

[65] Goyal, N., Vempala, S., and Xiao, Y., "Fourier pca and robust tensor decomposition," in *Proceedings of the 46th Annual ACM Symposium on Theory of Computing*, pp. 584–593, ACM, 2014.

[66] Guedon, O. and Milman, E., "Interpolating thin-shell and sharp large-deviation estimates for isotropic log-concave measures," *Geometric Functional Analysis*, vol. to appear, 2011.

[67] Guedon, O. and Rudelson, M., "L_p moments of random vectors via majorizing measures," *Advances in Mathematics*, vol. 208, pp. 798–823, 2007.

[68] Harshman, R. A., "Foundations of the PARAFAC procedure: models and conditions for an "explanatory" multimodal factor analysis," *UCLA Working Papers in Phonetics*, vol. 16, pp. 1–84, 1970.

[69] Håstad, J., "Some optimal inapproximability results," *J. ACM*, vol. 48, pp. 798–859, July 2001.

[70] Hastie, T., Tibshirani, R., Friedman, J., Hastie, T., Friedman, J., and Tibshirani, R., *The elements of statistical learning*, vol. 2. Springer, 2009.

[71] Hazan, E. and Krauthgamer, R., "How hard is it to approximate the best nash equilibrium?," *SIAM J. Comput.*, vol. 40, no. 1, pp. 79–91, 2011.

[72] Hillar, C. and Lim, L.-H., "Most tensor problems are NP-hard," *Journal of the ACM*, vol. 60, 2013.

[73] Hsu, D. and Kakade, S. M., "Learning mixtures of spherical Gaussians: moment methods and spectral decompositions," in *ITCS*, pp. 11–20, 2013.

[74] Hsu, D., Kakade, S. M., and Zhang, T., "A spectral algorithm for learning hidden Markov models," in *Proc. of COLT*, 2009.

[75] Hyvarinen, A., "Fast and robust fixed-point algorithms for independent component analysis," *Neural Networks, IEEE Transactions on*, vol. 10, no. 3, pp. 626–634, 1999.

[76] Hyvärinen, A., Karhunen, J., and Oja, E., *Independent Component Analysis.* Wiley, 2001.

[77] Hyvärinen, A. and Oja, E., "A fast fixed-point algorithm for independent component analysis," *Neural computation*, vol. 9, no. 7, pp. 1483–1492, 1997.

[78] Jerrum, M., "Large cliques elude the metropolis process," *Random Struct. Algorithms*, vol. 3, no. 4, pp. 347–360, 1992.

[79] Juels, A. and Peinado, M., "Hiding cliques for cryptographic security," *Des. Codes Cryptography*, vol. 20, no. 3, pp. 269–280, 2000.

[80] Jutten, C. and Herault, J., "Blind separation of sources, part i: An adaptive algorithm based on neuromimetic architecture," *Signal Processing*, vol. 24, no. 1, pp. 1 – 10, 1991.

[81] Kalai, A. T., Moitra, A., and Valiant, G., "Efficiently learning mixtures of two Gaussians," in *Proceedings of the 42nd ACM symposium on Theory of computing*, pp. 553–562, ACM, 2010.

[82] Kannan, R. and Vempala, S., *Spectral algorithms*. Now Publishers Inc, 2009.

[83] Karp, R., "Probabilistic analysis of graph-theoretic algorithms," in *Proceedings of Computer Science and Statistics 12th Annual Symposium on the Interface*, p. 173, 1979.

[84] Klivans, A. R., Long, P. M., and Tang, A. K., "Baum's algorithm learns intersections of halfspaces with respect to log-concave distributions," in *APPROX-RANDOM*, pp. 588–600, 2009.

[85] Klivans, A. R., O'Donnell, R., and Servedio, R. A., "Learning geometric concepts via Gaussian surface area," in *FOCS*, pp. 541–550, 2008.

[86] Kolda, T. G. and Bader, B. W., "Tensor decompositions and applications," *SIAM Rev.*, vol. 51, pp. 455–500, Aug. 2009.

[87] Kolda, T. G. and Mayo, J. R., "Shifted power method for computing tensor eigenpairs," *SIAM Journal on Matrix Analysis and Applications*, vol. 32, no. 4, pp. 1095–1124, 2011.

[88] Kreutz-Delgado, K., Murray, J. F., Rao, B. D., Engan, K., Lee, T.-W., and Sejnowski, T. J., "Dictionary learning algorithms for sparse representation," *Neural computation*, vol. 15, no. 2, pp. 349–396, 2003.

[89] Kruskal, J. B., "Three-way arrays: rank and uniqueness of trilinear decompositions, with application to arithmetic complexity and statistics," *Linear algebra and its applications*, vol. 18, no. 2, pp. 95–138, 1977.

[90] Kucera, L., "Expected complexity of graph partitioning problems," *Discrete Applied Mathematics*, vol. 57, no. 2-3, pp. 193–212, 1995.

[91] Lacoume, J.-L. and Ruiz, P., "Separation of independent sources from correlated inputs," *IEEE Transactions on Signal Processing*, vol. 40, no. 12, pp. 3074–3078, 1992.

[92] Lang, S., *Complex analysis*, vol. 103. Springer, 1998.

[93] Lang, S., *Undergraduate algebra*. Springer Verlag, 2005.

[94] Lee, J. A. and Verleysen, M., *Nonlinear dimensionality reduction*. Springer, 2007.

[95] Leurgans, S. E., Ross, R. T., and Abel, R. B., "A decomposition for 3-way arrays," *SIAM Journal on Matrix Analysis and Applications*, vol. 14, pp. 1064–1083, 1993.

[96] Lim, L.-h., "Singular values and eigenvalues of tensors: a variational approach," in *in CAMAP2005: 1st IEEE International Workshop on Computational Advances in Multi-Sensor Adaptive Processing*, Citeseer.

[97] Lovász, L. and Vempala, S., "The geometry of logconcave functions and sampling algorithms," *Random Struct. Algorithms*, vol. 30, no. 3, pp. 307–358, 2007.

[98] Marcinkiewicz, J., "Sur une propriété de la loi de Gauss," *Mathematische Zeitschrift*, vol. 44, no. 1, pp. 612–618, 1939.

[99] McSherry, F., "Spectral partitioning of random graphs," in *FOCS*, pp. 529–537, 2001.

[100] Minder, L. and Vilenchik, D., "Small clique detection and approximate nash equilibria," *Lecture Notes in Computer Science*, vol. 5687, pp. 673–685, 2009.

[101] Moitra, A. and Valiant, G., "Settling the polynomial learnability of mixtures of Gaussians," in *Foundations of Computer Science (FOCS), 2010 51st Annual IEEE Symposium on*, pp. 93–102, IEEE, 2010.

[102] Mossel, E., O'Donnell, R., and Oleszkiewicz, K., "Noise stability of functions with low influences: Invariance and optimality," *Annals of Math.*, vol. 171, pp. 295–341, 2010.

[103] Mossel, E., O'Donnell, R., and Servedio, R. A., "Learning functions of k relevant variables," *Journal of Computer and System Sciences*, vol. 69, pp. 421–434, 2004.

[104] Mossel, E. and Roch, S., "Learning nonsingular phylogenies and hidden markov models," in *STOC*, pp. 366–375, 2005.

[105] Motwani, R. and Raghavan, P., *Randomized algorithms*. Chapman & Hall/CRC, 2010.

[106] Ngiam, J., Chen, Z., Chia, D., Koh, P. W., Le, Q. V., and Ng, A. Y., "Tiled convolutional neural networks," in *Advances in Neural Information Processing Systems*, pp. 1279–1287, 2010.

[107] Nguyen, P. Q. and Regev, O., "Learning a parallelepiped: Cryptanalysis of GGH and NTRU signatures," *J. Cryptology*, vol. 22, no. 2, pp. 139–160, 2009.

[108] Olshausen, B. A. and others, "Emergence of simple-cell receptive field properties by learning a sparse code for natural images," *Nature*, vol. 381, no. 6583, pp. 607–609, 1996.

[109] P. M. Gruber, J. M. W. e., *Handbook of convex geometry. Vol. A. B.* North-Holland, Amsterdam, 1993.

[110] Pearson, K., "Contributions to the mathematical theory of evolution," *Philosophical Transactions of the Royal Society of London. A*, pp. 71–110, 1894.

[111] Pearson, K., "On lines and planes of closest fit to systems of points in space," *The London, Edinburgh, and Dublin Philosophical Magazine and Journal of Science*, vol. 2, no. 11, pp. 559–572, 1901.

[112] Press, W. H., Teukolsky, S. A., Vetterling, W. T., and Flannery, B. P., *Numerical recipes 3rd edition: The art of scientific computing*. Cambridge university press, 2007.

[113] Qi, L., "Eigenvalues of a real supersymmetric tensor," *Journal of Symbolic Computation*, vol. 40, no. 6, pp. 1302–1324, 2005.

[114] Rudelson, M., "Random vectors in the isotropic position," *J. of Functional Analysis*, vol. 164, pp. 60–72, 1999.

[115] Sanjeev, A. and Kannan, R., "Learning mixtures of arbitrary Gaussians," in *Proceedings of the thirty-third annual ACM symposium on Theory of computing*, pp. 247–257, ACM, 2001.

[116] Schwartz, J., "Fast probabilistic algorithms for verification of polynomial identities," *Journal of the ACM*, vol. 27, pp. 701–717, 1980.

[117] Shalvi, O. and Weinstein, E., "New criteria for blind deconvolution of nonminimum phase systems (channels)," *IEEE Transactions on Information Theory*, vol. 36, no. 2, pp. 312–321, 1990.

[118] Shiryaev, A., *Probability*. New York, NY: Springer-Verlag, 1995.

[119] Smilde, A., Bro, R., and Geladi, P., *Multi-way analysis: applications in the chemical sciences*. Wiley. com, 2005.

[120] Spielman, D. A., Wang, H., and Wright, J., "Exact recovery of sparsely-used dictionaries," in *Proceedings of the Twenty-Third international joint conference on Artificial Intelligence*, pp. 3087–3090, AAAI Press, 2013.

[121] Srivastava, N. and Vershynin, R., "Covariance estimates for distributions with $2 + \epsilon$ moments," *Annals of Probability*, vol. to appear (arXiv:1106.2775), 2011.

[122] Steele, J. M., *The Cauchy–Schwarz Master Class*. Cambridge University Press, 2004.

[123] Stewart, G. W. and Sun, J.-g., *Matrix perturbation theory*. Academic press, 1990.

[124] Thompson, R. C., "Principal submatrices IX: Interlacing inequalities for singular values of submatrices," *Linear Algebra and its Applications*, vol. 5, pp. 1–12, 1972.

[125] Vempala, S., "Learning convex concepts from Gaussian distributions with pca," in *FOCS*, pp. 541–550, 2010.

[126] Vempala, S., "A random sampling based algorithm for learning the intersection of half-spaces," *JACM*, vol. 57, pp. 32:1–32:14, 2010.

[127] Vempala, S. and Wang, G., "A spectral algorithm for learning mixture models," *Journal of Computer and System Sciences*, vol. 68, no. 4, pp. 841–860, 2004.

[128] Vempala, S. and Xiao, Y., "Complexity of learning subspace juntas and ica," in *Signals, Systems and Computers, 2013 Asilomar Conference on*, pp. 320–324, IEEE, 2013.

[129] Vempala, S. S. and Xiao, Y., "Structure from local optima: Learning subspace juntas via higher order PCA," *CoRR*, vol. abs/1108.3329, 2011.

[130] Vershynin, R., "Introduction to the non-asymptotic analysis of random matrices," in *Compressed Sensing, Theory and Applications* (Eldar, Y. and Kutyniok, G., eds.), pp. 210–268, Oxford: Cambridge University Press, 2010.

[131] Vershynin, R., "How close is the sample covariance matrix to the actual covariance matrix?," *Journal of Theoretical Probability*, vol. 25, no. 3, pp. 655–686, 2012.

[132] Wilkinson, J. H., *The algebraic eigenvalue problem*, vol. 155. Oxford Univ Press, 1965.